

A PARALLEL PRECONDITIONING STRATEGY FOR EFFICIENT TRANSISTOR-LEVEL CIRCUIT SIMULATION*

E. G. BOMAN, D. M. DAY, R. J. HOEKSTRA, E. R. KEITER, H. K. THORNQUIST
SANDIA NATIONAL LABORATORIES
ALBUQUERQUE, NM 87185-0316

Abstract. A parallel computing approach for large-scale SPICE-accurate circuit simulation is described that is based on a new preconditioned iterative solver. The preconditioner involves the Dulmage-Mendelsohn decomposition and hypergraph partitioning. Our parallel implementation makes use of a mixed load balance, employing a different parallel partition for the matrix load and solve. Based on message-passing, our simulator was designed for large parallel computers, but here good parallel speedup in modern multi-core environments is demonstrated.

Traditional circuit simulation, originally made popular by the Berkeley SPICE program, relies on sparse direct matrix solver methods and does not scale well beyond tens of thousands of unknowns. Linear systems resulting from the simulation of circuits are sparse, typically have heterogeneous non-symmetric structure, and are often ill-conditioned.

Parallel circuit simulation has been investigated previously, including Basermann [2] who used a Schur-complement based preconditioner, and Cheng [12] uses factorizations as a preconditioner. However, for conventional transient full-system simulation iterative matrix solvers have yet to be widely used.

The new solver, BTF, is compared to a serial direct solver (KLU), a parallel direct solver (SuperLU_DIST) and standard iterative solvers ([6]) on a set of circuit test problems. KLU is a serial sparse direct linear solver developed specifically for circuit simulation [14]. SuperLU_DIST is a general purpose (not circuit-specific) distributed memory sparse direct linear solver [9]. We also tried PARDISO [13], a sparse solver for multi-core processors, but excluded it from the comparison after disappointing preliminary results. In the results presented KLU reliably solves all the test circuits, and is considered the baseline for comparison.

The third comparison method is a general purpose distributed memory iterative linear solver, combining four steps. The first steps are Singleton Removal [2], and graph partitioning (ParMETIS [8]) on the resulting symmetrized graph to reduce communication. Step three is a local fill-reducing ordering (AMD [1]) on the subdomain. And step four is either a complete or incomplete LU factorization. Note that domain decomposition using an exact subdomain solve and BTF differ only in reordering and partitioning.

Traditional circuit simulation is based on the differential algebraic equation (DAE):

$$(0.1) \quad f(x(t)) + \frac{dq(x(t))}{dt} = b(t)$$

Simulation of this transient equation results in linear systems of the form:

$$(0.2) \quad (G + Q/\delta t)\delta x = (b - f)/\delta t$$

involving the conductance matrix $G(t) = \frac{df}{dx}(x(t))$, and the capacitance matrix $Q(t) = \frac{dq}{dx}(x(t))$.

The system of equations (0.1) is solved by implicit methods. In the initial DC operating point (DCOP) problem, the q terms are not present, so equation (0.1) is reduced to the nonlinear equation $f(x) = 0$, and the linear system is simplified to:

$$(0.3) \quad G \delta x = -f(x).$$

Our motivation for the new linear solution strategy is the observation that the conductance matrix G is always reducible, and often may be permuted to a unique block triangular form with tiny diagonal blocks. The challenge is that the Jacobian matrix for the transient DAE is not reducible. Our contribution is a preconditioner for the transient problem, equation (0.1), that respects the block structure of G .

The Dulmage-Mendelsohn decomposition [4] of the initial G is determined once and only once. First a maximum matching permutation determines a matrix with a zero-free diagonal, and second a

* Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

topological sort finds the strongly-connected components of the associated directed graph, resulting in the permutation to block triangular form (BTF). For the remainder of DCOP, equation (0.3), as well as for the transient problem, equation (0.2), the unknowns within each block are always on the same subdomain. This is accomplished by partitioning a coarsened graph, in which a vertex corresponds to a block of the original G . An exact subdomain solve using KLU is used.

A parallel Dulmage-Mendelsohn decomposition is not available at this time, and would involve two steps. The first step, a bipartite matching, is the most difficult [10], but in all our experiments, Singleton Removal determines matrices without zeros on the diagonals. The second step finds strongly connected components and recent work shows this can be done in parallel [7, 11].

Hypergraph partitioning has shown a lot of promise [3, 5], in part because it relies upon a more accurate metrics for measuring parallel communication costs, and will give an exact estimate of the costs required for a matrix-vector multiply, the main communication expense for parallel GMRES. In the hypergraph partitioner, each row corresponds to a vertex, and each column represents a hyperedge. The result of the partitioning is an assignment of rows to processors.

The new solver reduces the total simulation time by up to a factor eight compared to the serial solver KLU on a 16-core computer. The BTF preconditioner works well whenever the irreducible blocks (strongly connected components) are small, e.g. for seven out of the ten test circuits, but future modifications are needed for G with a large irreducible block.

Key words. parallel simulation, circuit simulation, iterative matrix solvers, preconditioner, hypergraph partitioning, multi-core

REFERENCES

- [1] P. Amestoy, T.Davis, and I.Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Appl.*, 17(4):886–905, 1996.
- [2] A. Basermann, U. Jaekel, and M. Nordhausen. Parallel iterative solvers for sparse linear systems in circuit simulation. *Fut. Gen.. Comput. Sys.*, 21(8):1275–1284, 2005.
- [3] Ü. Çatalyürek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Trans. Parallel Dist. Systems*, 10(7):673–693, 1999.
- [4] T. A. Davis. *Direct Methods for Sparse Linear System*. SIAM, 2006.
- [5] K.D. Devine, E.G. Boman, R.T. Heaphy, R.H. Bisseling, and U.V. Catalyurek. Parallel hypergraph partitioning for scientific computing. In *Proc. of 20th International Parallel and Distributed Processing Symposium (IPDPS'06)*. IEEE, 2006.
- [6] M. Heroux et al. An overview of the Trilinos project. *ACM TOMS*, 31(3):397–423, 2005.
- [7] L. Fleischer, B. Hendrickson, and A. Pinar. On identifying strongly connected components in parallel. In *Proc. of IPDPS'00, LNCS 1800*, pages 505–511. Springer, 2000.
- [8] V. Kumar K. Schloegel, G. Karypis. Parallel static and dynamic multi-constraint graph partitioning. *Concurrency and Computation: Practice and Experience*, 14(3):219–240, 2002.
- [9] X. S. Li and J. W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Math. Softw.*, 29(2):110–140, 2003.
- [10] F. Manne and R. H. Bisseling. A parallel approximation algorithm for the weighted maximum matching problem. In *Proc. of PMAA'08, LNCS 4967*, pages 708–717. Springer, Berlin, 2008.
- [11] W. McLendon, B. Hendrickson, S. Plimpton, and L. Rauchwerger. Finding strongly connected components in distributed graphs. *J. of Parallel and Distributed Computing*, 65:901–910, 2005.
- [12] He Peng and Chung-Kuan Cheng. Parallel transistor level circuit simulation using domain decomposition methods. In *Proceedings of the 2009 Conference on Asia and South Pacific Design Automation*, pages 397–402. IEEE Press, 2009.
- [13] O. Schenk and K. Gärtner. Solving unsymmetric sparse systems of linear equations with PARDISO. *Fut. Gen.. Comput. Sys.*, 20(3):475–487, 2005.
- [14] K. Stanley and T. Davis. KLU: a ‘Clark Kent’ sparse LU factorization algorithm for circuit matrices. In *SIAM Conference on Parallel Processing for Scientific Computing (PP04)*, 2004.