

Parallel Circuit Simulation

Alexander Hayman

In the News

LTspice IV features multi-threaded solvers designed to better utilize current multicore processors. Also included are new SPARSE matrix solvers that deploy assembly language in order to approach the theoretical flop limit of current FPUs (floating point units). LTspice IV speeds up the simulation speed of medium- to large-sized circuits by a factor of three on a quad core. According to Linear Technology, LTspice IV outperforms other commercially available SPICE programs. LTspice IV implements proprietary methods that efficiently parallelize tasks that would require as little as 5us to run single-threaded.

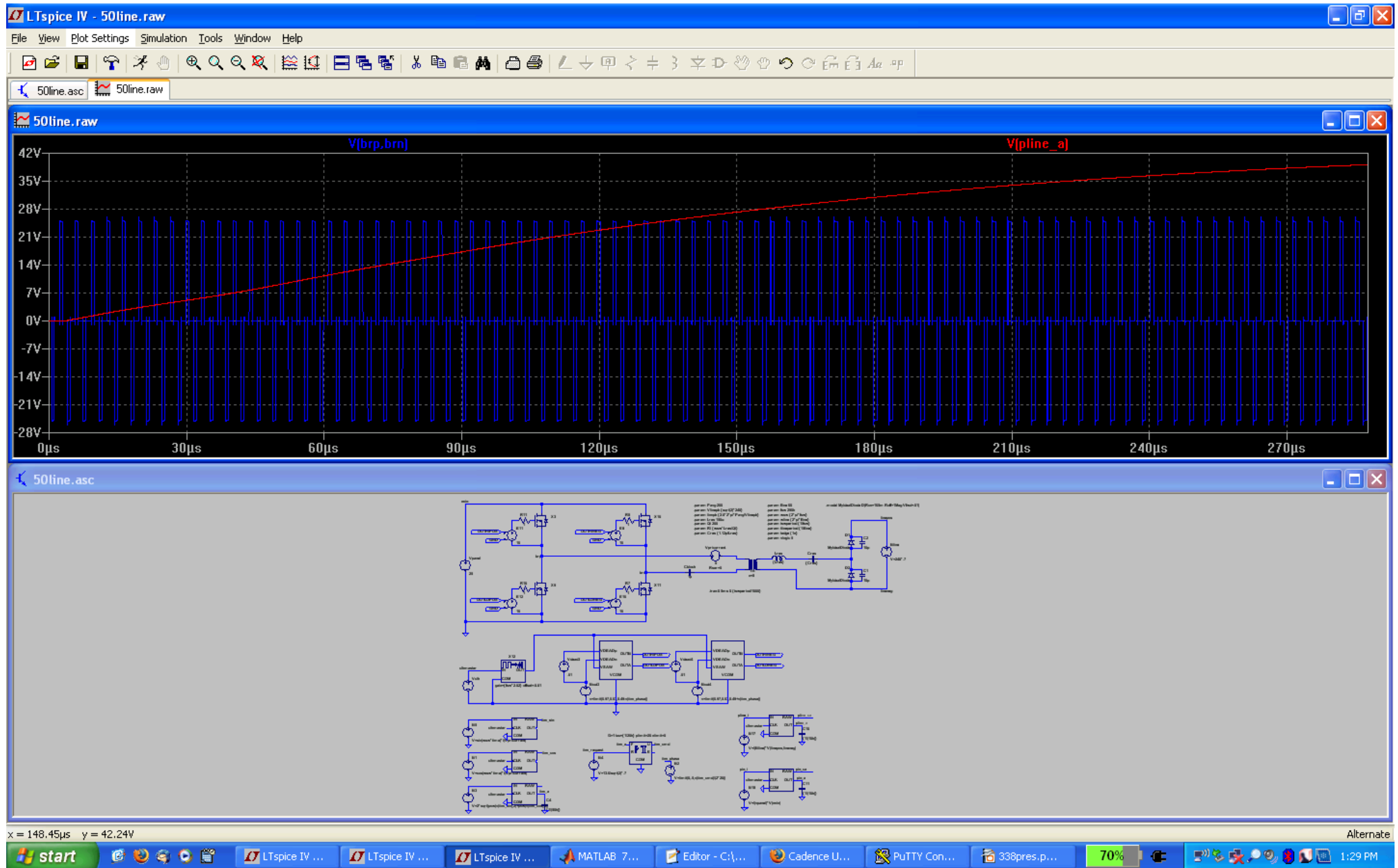
January 2009

Cadence Launches Virtuoso Accelerated Parallel Simulator for Circuits

“It is developed to solve large and complex analogue and mixed-signal designs across all process nodes. The Cadence MMSIM 7.1(multi-mode simulation solution) simulator consists of Cadence simulation technologies, parallel circuit solver and an architected engine that effectively harnesses the power of multiprocessing computing platforms. The circuit simulator is capable of delivering improved single-thread performance and scalable multi-thread performance.”

December 2008

LTSpice IV



Circuit Simulation Basics

- For a transient analysis, use numerical integration to keep track of energy storage components.
- At each time step, use Newton's method to solve a non-linear highly sparse system.
- Newton's method involves solving $Ax=b$ multiple times for varying $\underline{b}$.
- This is most efficiently accomplished by finding LU factorization of A .

LU Factorization

- LU factorization of highly sparse matrices is the most computational intensive task for circuit simulators.
- LU factorization must be performed at each time step.

Kent LU

Sparse LU factorization techniques:

- Based on study by Tim Davis at UCSD
 - Will look at ways to parallelize KLU algorithm
 - This algorithm is specifically aimed at sparse matrices that are factorized during circuit simulations.
- licable to 3-D heat flow, and struts/joints./joints.

KLU

- Tim Davis at UF developed an LU factorization algorithm specifically aimed at solving circuit matrices.
- KLU algorithm is a series of sequential steps:
 - Preprocessing
 - Block Triangular Form permutation (BTF)
 - Approximate Minimum Degree (permute blocks to minimize fill)
 - LU factorization (Gilbert/Peierls algorithm)
 - Solve

BTF

- Permutation to block upper triangular form.
- Two sequential steps:
 - Find the maximum transversal (column perm. to minimize zeros on the diagonal)
 - Find the strong components (minimize the size of the blocks on the diagonal)
- Both steps involve a depth first search.
- AMD and LU factorization also involve depth first search.
- Parallelize depth first search?

Parallel Depth First Search

- Use multiple computers to search different spaces of a tree.
- When first solution is found, kill all processes, and move to next step.
- Difficult in practice, but easy to implement a Las Vegas algorithm (randomly select which branches in tree to iterate through).
- Only effective if there is some backtracking during the search.

Results

- Implemented a randomized maximum transversal algorithm by making modifications to SuiteSparse published by Tim Davis.
- Slowed down speed of function for nearly all circuit description matrices (from Tim Davis sparse matrix database).
- Indicates that there is not enough backtracking in default case to justify cost of generating random permutations.

Future Work

- Parallelization of depth first search for other components of the KLU algorithm may improve speed.
- Should also test whether random selection of a branching rule is more effective than random selection of a branch.
 - Eliminates need to generate random permutations.
 - Still increases the breadth of the search.