

Fast Algorithm for Clock Grid Simulation

P. Sotiriadis¹, O. Franza², D.W. Bailey², B. Calhoun¹, D. Lin², and A. Chandrakasan¹

¹ *Massachusetts Institute of Technology, Cambridge, MA 02139*

² *Alpha Development Group, Compaq, Shrewsbury, MA 01545,
currently with Intel Corporation, Shrewsbury, MA 01545*

pps@mit.edu, Olivier.Franza@intel.com

Abstract

A fast and reliable design tool for real-time simulation of clock grids is presented here. Its implementation is orders of magnitude faster than SPICE, which makes it well-suited for the analysis of a multitude of design options early in a project. The mathematical formulation fully models a regular RLC clock grid with arbitrarily located loads and input sources. The algorithm is validated using numerical data and SPICE simulated skew numbers from the Alpha 21364 microprocessor.

1. Introduction

Clock grids have several appealing characteristics that make them an appropriate choice for digital integrated circuit designs. They are tolerant to process, voltage, temperature, and load variations, have design convenience for chip circuitry, and have deterministic skew that can be neutralized by incorporating the RC skew into the timing tools. The sum of these advantages results in a clock design that has a compelling time-to-market advantage because it can easily accommodate late design changes in underlying chip circuitry. This outlines the reasons many high-end microprocessors have used grids as part of their clock distribution network hierarchy [1]-[4].

The shortcomings of clock grids, however, are well known by those who have studied their advantages. First, they use more power than an optimized approach [5], simply because there is generally more interconnect than for point-to-point routing. Second, the simulation time can be daunting. A modern microprocessor clock grid that covers its die requires a huge RLC network with non-uniform capacitive loads. Using SPICE to model this type of grid can take an impractically long time (i.e., more than a few days). Considering that SPICE simulation time goes as the cube of the number of elements, one can easily see that formerly applicable techniques may not scale to next-generation designs.

Hence, if power concerns can be overcome, simulation time is a major issue possibly blocking the adoption of a clock grid. In the incipient stages of the design, before the floorplan is stable, quick turnaround time is essential for a clock CAD tool. During this early design phase, exper-

imental comparisons are necessary, and major implementation and layout changes are common. Since the loads are not well known, the usefulness of SPICE-level accuracy is questionable. There is, consequently, a design niche for a clock grid simulation CAD tool that can quickly solve arbitrarily large grids with reasonable accuracy.

Previously published methods for fast grid solutions do not meet all the requirements of this early design phase. Asymptotic Wave Expansion [6]-[7] has proven to be a good choice for simple RC grids, but becomes unstable once inductance is taken into account. Model Order Reduction [8],[9] is a method leading to accurate results, but has a long setup time which is not suited to frequent design changes. Such a method is more appropriate for the analysis of a given structure throughout different operating conditions, not the analysis of a wide range of design options for various capacitive load patterns. Other explicit fast solution methods [10]-[11] are available, but are prone to more complexity, instability, or roundoff errors from alternating between continuous and discrete modeling.

The algorithm described in this paper solves for clock skew on an RLC network with linear simulation time. It assumes a rectangular grid element with uniform R and L interconnect parameters, but arbitrary C (gate and interconnect) load and placement. Clock grid drivers are approximated by Thevenin sources with time-varying resistances, thus modeling input driver skew due to variations in the clock distribution network. Driver placement is also arbitrary on the grid. Although the grid elements must be rectangular, the method allows missing grid points, so that a huge variety of grid shapes is permitted. Overall, the method is quite general and useful for clock grid design.

2. Method Presentation

Assume in the following a rectangular, regular, and uniform clock grid made of $N \times M$ nodes. Every node V_{ij} on the grid is connected to its horizontal and vertical neighbors on the grid with an RLC-modeled wire. In addition, a source and a load can be attached to it. Figure 1 shows the model of node V_{ij} and its nearest neighbors. All interconnects are assumed to have the same R and L . The term C_{ij} is the capacitance to ground of node (i, j) , which includes both the interconnect capacitance and any load

capacitance from the gates driven by the clock grid. $r_{ij}(t)$ is the time-dependent equivalent resistance of the driver (if any) connected to node (i, j) . If there is no driver, then $r_{ij} \rightarrow \infty$. The voltage source $u_{ij}(t)$ is arbitrary and

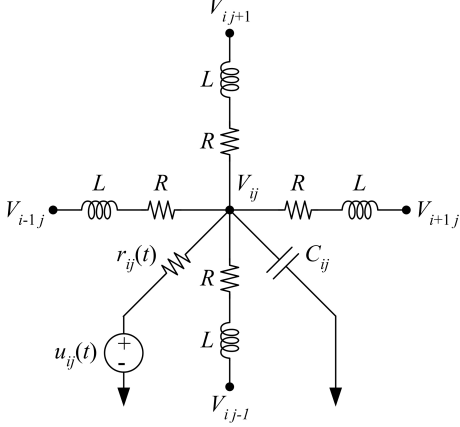


Figure 1. Internal grid element model

independent from all other nodes. For boundary and corner nodes of the grid, the model is similar but with only three and two neighboring nodes, respectively. Applying Kirchhoff's current law in Laplace space to internal node (i, j) yields

$$sC_{ij}V_{ij} = \frac{1}{Ls + R}(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} - 4V_{ij}) + \frac{1}{R_{ij}}(U_{ij} - V_{ij}) \quad (1)$$

where U_{ij} and R_{ij} are the Laplace transforms of the voltage source and the driver resistance, respectively. To simplify the notation, operator \mathcal{D} is defined as follows:

$$\mathcal{D}V_{ij} = V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} - 4V_{ij} \quad (2)$$

for any internal node. The expression of \mathcal{D} is similar for side-boundary nodes (with 4 terms only), and for the corner nodes (with 3 terms only). Equation (1) becomes

$$sC_{ij}V_{ij} = \frac{1}{Ls + R}\mathcal{D}V_{ij} + \frac{1}{R_{ij}}(U_{ij} - V_{ij}) \quad (3)$$

or, in the time-domain, a differential equation of the second order,

$$LC_{ij}\ddot{V}_{ij} + RC_{ij}\dot{V}_{ij} = \mathcal{D}V_{ij} + \frac{L}{r_{ij}}(\dot{u}_{ij} - \dot{V}_{ij}) + \frac{R}{r_{ij}}(u_{ij} - V_{ij}), \quad (4)$$

$\forall(i, j) \in [1, N] \times [1, M]$. Note that this equation would be of the third order if the grid was nonuniform in terms of L and R . Defining the new set of variables $\{W_{ij}\}_{(i,j) \in [1,N] \times [1,M]}$ such that

$$\dot{V}_{ij} = W_{ij} + \frac{1}{r_{ij}C_{ij}}u_{ij} \quad (5)$$

and substituting Equation (5) into Equation (4) leads to the following matrix system:

$$\begin{bmatrix} \dot{V}_{ij} \\ \dot{W}_{ij} \end{bmatrix} = \begin{bmatrix} W_{ij} \\ \mathcal{T}V_{ij} + \mathcal{S}W_{ij} \end{bmatrix} + \begin{bmatrix} q_{ij} \\ -q_{ij}^2 \end{bmatrix} u_{ij} \quad (6)$$

where $\mathcal{T} = p_{ij}\mathcal{D} - \frac{R}{L}q_{ij}$, $\mathcal{S} = -\frac{R}{L} - q_{ij}$, $p_{ij} = \frac{1}{LC_{ij}}$, $q_{ij} = \frac{1}{r_{ij}C_{ij}}$, and P and Q_1 are their associated matrices, $\forall(i, j) \in [1, N] \times [1, M]$. Finally, in full-operator form, Equation (6) can be written as a $2N \times M$ system:

$$\dot{Y} = \mathcal{A}Y + u \cdot Z \quad (7)$$

where $Y = \begin{bmatrix} V \\ W \end{bmatrix}$, operator $\mathcal{A} = \begin{bmatrix} 0 & \mathcal{I} \\ \mathcal{T} & \mathcal{S} \end{bmatrix}$ is linear in Y (where \mathcal{I} is the identity operator), and $Z = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$ with $Q_2 = \{-q_{ij}^2\}_{(i,j) \in [1,N] \times [1,M]}$. To solve Equation (7), explicit methods like Euler's cannot be applied, as they do not meet the following requirements: unconditional stability, very low startup time, very low memory requirement. A method with such properties is the implicit one that is used in SPICE, the second-order Runge-Kutta method, and can be defined for variable Y as

$$Y_k = Y_{k-1} + \frac{h}{2}(\dot{Y}_k + \dot{Y}_{k-1}) \quad (8)$$

where h is the time step, $Y_k = Y(t_k)$, and $t_k = k \cdot h$, $\forall k \in \mathbb{N}$. Discretizing Equation (7) accordingly and including it into Equation (8) finally yields

$$\left(\mathcal{I} - \frac{h}{2}\mathcal{A}\right)Y_k = \left(\mathcal{I} + \frac{h}{2}\mathcal{A}\right)Y_{k-1} + \frac{h}{2}u(t_k) \cdot Z(t_k) + \frac{h}{2}u(t_{k-1}) \cdot Z(t_{k-1}). \quad (9)$$

Solving Equation (9) is computationally expensive due to the inversion of linear operator $\mathcal{B} = \mathcal{I} - \frac{h}{2}\mathcal{A}$. However, by setting the time step such that $\frac{h}{2} \|\mathcal{A}\| \leq 1$ for a given norm, the inverse of \mathcal{B} can be approximated by a Taylor expansion of the form

$$\text{inv}(\mathcal{B}) \cong I + \sum_{n \in \mathbb{N}^*} \frac{h^n}{2^n} \mathcal{A}^n. \quad (10)$$

An expansion to the fourth order showed to be sufficient. If the *infinity*-norm is used, it can be shown, using Equation (7), that

$$\|\mathcal{A}\| \leq \max \left\{ 1, \max \left\{ 8p_{ij} + \frac{R}{L} |q_{ij}| + |s_{ij}| \right\} \right\}, \quad (11)$$

which allows us to determine an upper bound for h :

$$h \leq \frac{2}{\max \left\{ 1, \max \left\{ \frac{8}{LC_{ij}} + \frac{R}{Lr_{ij}C_{ij}} + \frac{R}{L} + \frac{1}{r_{ij}C_{ij}} \right\} \right\}}. \quad (12)$$

3. Numerical Validation

After successful validation of the method on canonical cases, real data were used from the Alpha 21364 microprocessor [1]. The 21364 has two levels of clock grid, a coarse one (called NCLK), covering most of the die, and a finer one, two levels of inversions after NCLK, made of local independent clock domains associated with different logical units throughout the die. Presented here are the results obtained for one of the local domains, called Z1CLK, and for NCLK.

For Z1CLK, the total number of grid points for the fast algorithm is $92 \times 162 = 14,904$, leading to about 95,000 elements in SPICE. The non-uniform load distribution and 69 driver locations are shown in Figure 2. The simulated skew results are presented in Figures 3 and 4 and are in good agreement. The difference in skew is less than 10% at all nodes. The SPICE simulation took 4 hours of CPU time on an ES40 AlphaServer and the fast algorithm less than 10 minutes on a Pentium 4 workstation.

For NCLK, the total number of grid points for the fast algorithm is $188 \times 266 = 50,008$, leading to over 200,000 elements in SPICE. The non-uniform load distribution and 320 driver locations are shown in Figure 5. The simulated skew results are presented in Figures 6 and 7. Skew values agree quite well throughout the grid. One area of discrepancy is noticeable near point (75, 150). In this region, the final NCLK grid was depopulated horizontally [1], which is not included in our simulation. The simulation took more than 12.5 hours with SPICE and about 30 minutes with the fast algorithm. This $25 \times$ speedup gets dramatically larger for bigger clock grids.

4. Conclusion

The fast algorithm has been presented here and validated numerically on existing data from the Alpha 21364 microprocessor. Its speed, versatility, accuracy, and ease of use make it an appropriate design tool for clock grid analysis, and more specifically for skew analysis. This algorithm can also, with little added complexity or CPU time, compute risetime, overshoot and undershoot. With slightly more effort, it can also include different wire parameters for orthogonal interconnect directions. Although not a replacement for SPICE for when extremely accurate simulations are needed, it allows the exploration of a vast number of clock grid options as well as a quick analysis of any last-minute floorplan change or major circuit redesign.

5. Acknowledgements

The authors thank Johann Nittman, Stephen Oxx, Horst Wagner, and Ted Equi for functional assistance and Duke Xanthopoulos for technical contributions.

6. References

- [1] T. Xanthopoulos, D.W. Bailey, A.K. Gangwar, M.K. Gowan, A.K. Jain, and B.K. Prewitt, "The design and analysis of the clock distribution network for a 1.2 GHz Alpha microprocessor," *Dig. Technical Papers Solid-State Circuits Conf.*, IEEE, Feb. 2001, pp. 402-403.
- [2] S. Tam, S. Rusu, U.N. Desai, R. Kim, J. Zhang, and I. Young, "Clock generation and distribution for the first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, IEEE, vol. 35, no. 11, pp. 1545-1552, Nov. 2000. See also S. Rusu and G. Singer, "The first IA-64 microprocessor," *ibid*, pp. 1539-1544.
- [3] D.A. Draper *et al.*, "An X86 microprocessor with multimedia extensions," *ISSCC 1997 Dig. Technical Papers*, IEEE, Feb. 1997, pp. 1726-1737.
- [4] P.J. Restle *et al.*, "A clock distribution network for microprocessors," *J. Solid-State Circuits*, IEEE, vol. 36, no. 5, pp. 792-799, May 2001.
- [5] P.J. Restle, A. Deutsch, "Designing the best clock distribution network," *1998 Symp. VLSI Circuits Dig. Technical Papers*, 1998, pp. 2-5.
- [6] L.T. Pillage and R.A. Rohrer, "Asymptotic waveform evaluation for timing analysis", *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on, Vol. 9 Issue 4, April 1990, pp. 352-366.
- [7] D.F. Anastasakis, N. Gopal, S.-Y. Kim, L.T. Pillage, "Enhancing the stability of asymptotic waveform evaluation for digital interconnect circuit applications", *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on, Vol. 13 Issue 6, June 1994 pp. 729-736.
- [8] L. Miguel Silveira, M. Kamon, and J. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures", *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, IEEE, Vol. 19 Issue 2, May 1996, pp. 283-288.
- [9] T.H. Chen and C. Chen, "Efficient Large-Scale Power Grid Analysis Based on Preconditioned Krylov-Subspace Iterative Methods", *Design Automation Conference*, Proceedings, 2001, pp. 559-562.
- [10] Y.M. Lee and C.C. Chem, "Power Grid Transient Simulation in Linear Time Based on Transmission-Line-Modeling Alternating-Direction-Implicit Method", *Computer Aided Design ICCAD*, IEEE/ACM International Conference on, 2001, pp. 75-80.
- [11] A.B. Nassif and C.D. Kozhaya, "Fast Power Grid Simulation", *DAC*, IEEE Proceedings 37th Conf. on Design Automation, Los Angeles, 2000, pp. 156-161.

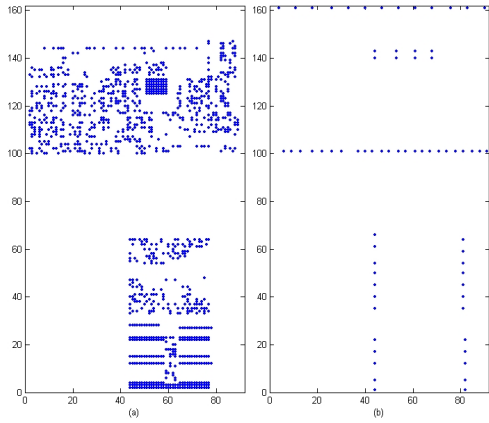


Figure 2. Z1CLK (a) load and (b) driver distributions

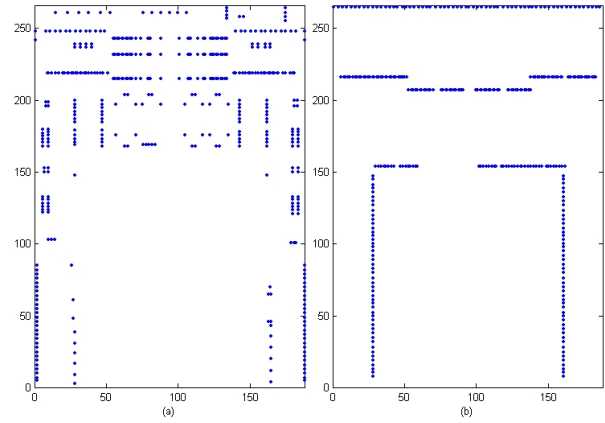


Figure 5. NCLK (a) load and (b) driver distributions

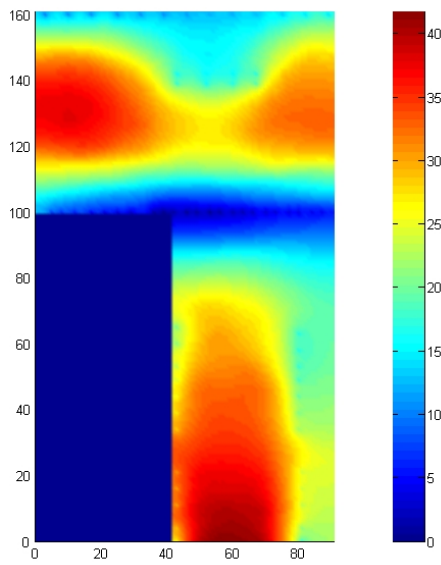


Figure 3. Z1CLK skew from SPICE

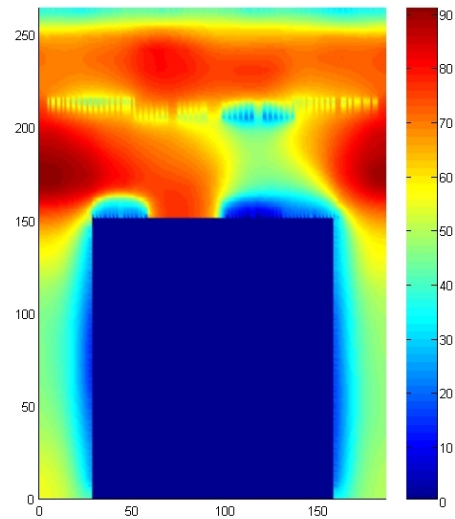


Figure 6. NCLK skew from SPICE

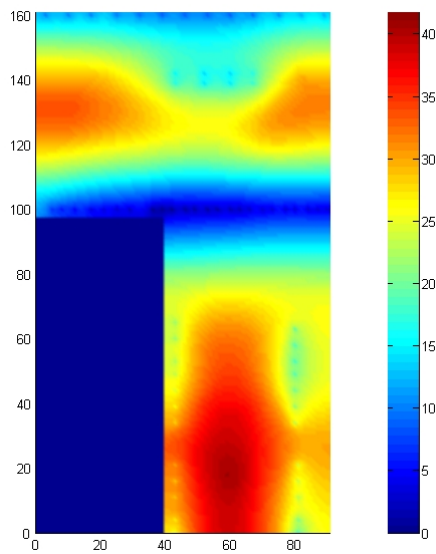


Figure 4. Z1CLK skew from the fast algorithm

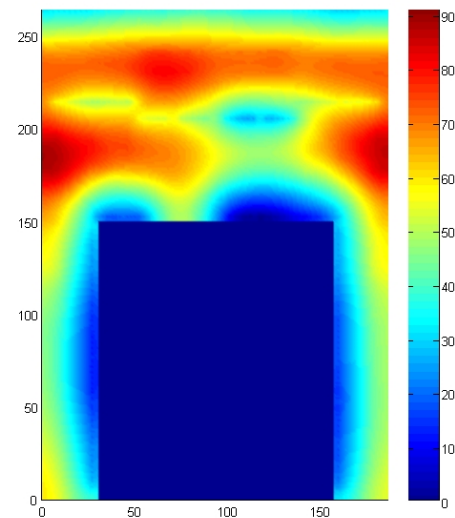


Figure 7. NCLK skew from the fast algorithm