

The Design and Implementation of PowerMill*

Charlie X. Huang[†] Bill Zhang[†] An-Chang Deng[†] Burkhard Swirski[‡]

Abstract

In this paper we discuss the design and implementation of the simulator PowerMill, a novel transistor level simulator for the simulation of current and power behavior in VLSI circuits. With a new transistor modeling technology and a versatile event driven simulation algorithm, PowerMill is capable of simulating detailed current behavior in modern deep-submicron CMOS circuits, including sophisticated circuitries such as exclusive-or gates and sense-amplifiers, with speed and capacity approaching conventional gate level simulators. The high accuracy and speed have made it possible for designers to study and verify detailed current behavior of large functional blocks or even an entire chip with a reasonable amount of CPU resources, making it a de facto industry standard for power simulation.

1 Introduction

Higher integration density, smaller device geometry, larger chip size, faster clock frequency, and the demand low power consumption have made power-related issues increasingly critical in VLSI circuits [1, 2]. For battery-operated devices, such as those in portable and hand-held applications, achieving minimum power consumption is usually the primary design goal. Controlling power consumption is also becoming an important design goal in designs that aren't battery-operated. This is because the excessive heat generated from high power consumption can seriously degrade chip performance and cause physical damage to the chip. It may also lead to increased packaging cost, which is a major cost factor for IC's. Electronmigration in metal lines and vias [3], due to high current densities, gives rise to chip failures in the field. Unusually high current spikes can lead to significant voltage fluctuations, both resistive and inductive, if the power distribution network is not properly designed. Such fluctuations in turn can lead to a wide variety of problems, from unintended logic transitions due to the lowered noise margin, to the slow down in localized portions of a circuit — which will often cause timing errors — as the effective supply voltage is reduced.

As process and design technology advance to make IC's larger and faster, these problems will only become worse. Unfortunately, many of them are transient in nature, and depend critically on the physical charac-

teristics of an IC, which will not become available until very late in the design cycle. Some of them, such as electronmigration, will not manifest themselves until after extended use in the field. It is clear that an effective and efficient analysis tool is needed to help designers address these problems.

2 Existing approaches

Probabilistic approach [4, 5, 6, 7] for the estimation of power consumption has attracted a lot of attention. Instead of simulating a circuit, this approach computes and propagates the *probability* for a node to change its logic state. Since it doesn't require a lengthy vector set, it can be quite fast. Nevertheless, its scope is confined, to a large extent, to the power consumption arising from the *dynamic* charging and discharging of capacitors. While it can be argued that this is generally the dominant component for power consumption in logic circuits, its dominance for memory circuits, NMOS design, and non-digital circuits is questionable at best.

The idea to compute power consumption based on logic changes on nodes is also used in another class of power simulators, which works by keeping track of node toggles using existing gate level or switch level simulators [8]. It suffers from the same limitation as the probabilistic approach in that non-capacitive power consumption is generally ignored. Although recent refinements have attempted to approximate some short-circuit power consumption, due to finite rise/fall times, such approximations are usually pre-built into a gate level library and is limited to some very specific design styles, and usually cannot handle full custom designs and memory designs.

3 PowerMill's approach

EPIC Design Technology's PowerMill is a highly accurate and efficient transistor level simulator. Being a transistor level tool, it is capable of handling the full spectrum of digital MOS circuits and many analog circuits. Its novel piecewise linear transistor model captures transistor characteristics in a table, which provides high accuracy with minimum model evaluation overhead. A unique event driven algorithm is employed to exploit circuit latency to achieve a speed and capacity level approaching that of logic simulators. Furthermore, in contrast to most switch and gate level simulators, events are determined in terms of small voltage changes, as opposed to logic transitions, making it possible to handle non-digital behaviors in an accurate manner.

*PowerMill is a trademark of EPIC Design Technology, Inc.. Some subject matters in this paper are covered by one or more pending U.S. patents.

[†]EPIC Design Technology Inc., Santa Clara, CA, U.S.A.

[‡]ITT-Intermetall, Freiburg, Germany

3.1 Piecewise linear modeling

Most timing simulators [9] use the resistor model to model transistors. In this model, a transistor is converted into a resistor, in series with a switch, between the drain and source terminals. The switch is closed if the logic state of the gate terminal is high/low for an NMOS/PMOS transistor, and opened otherwise (Figure 1.) The value of the resistor is typically set to that which yields the same delay charging/discharging a capacitor, from zero to V_{dd} , as would the transistor. Some more sophisticated tools also use the slew rate of the gate terminal to modify the resistance.

While this type of approach offers the maximum efficiency, compared to a full-blown SPICE-like analytical model, it is not capable of providing the required generality and accuracy that we had in mind. In designs with tight feedbacks, such as latches, memory cells, and typical domino logic gates, and in cases when the voltage waveforms of the terminals deviate too much from being a line trajectory between 0 and V_{dd} , this model simply cannot provide satisfactory accuracy.

To get the requisite generality and accuracy, we decided to revert to the companion models used by circuit simulators. In such a model, a transistor is represented as a current source, a transconductor, and a resistor, the values of them dependent on the voltages on all three terminals (Figure 2). Mathematically, this model is arrived at by expanding a nonlinear device model equation into a Taylor series near the operating point (v_{gs_0}, v_{ds_0}) , which is in turn truncated into a linear equation:

$$\begin{aligned} i_{ds} &= f(v_{gs}, v_{ds}) \\ &\approx i_0 + \frac{\partial f}{\partial v_{gs}} \Big|_{v_{ds}=v_{ds_0}} (v_{gs} - v_{gs_0}) \\ &\quad + \frac{\partial f}{\partial v_{ds}} \Big|_{v_{gs}=v_{gs_0}} (v_{ds} - v_{ds_0}). \end{aligned}$$

This approximation is always valid for a continuous function, and accurate within the neighborhood of the operating point. The only difference between SPICE and PowerMill is that, in SPICE, model equations and partial derivatives are expressed in terms of complicated analytical equations, and must be evaluated whenever the operating point changes, and in PowerMill they are precomputed and stored in table forms that can be quickly looked up. The loss of accuracy because of this approximation is very small, yet the gain in efficiency is tremendous. This is borne out by the fact that model evaluation accounts for a very small portion of the total CPU time. The generality of the model has allowed the use of PowerMill on deep sub-micron designs without appreciable loss of accuracy.

The variable v_{gs} in the discussion above should really have been $v_{gs_{effective}}$, which is the difference between the gate voltage and the source voltage, less the threshold voltage v_{th} . For any transistors that do not have their source terminal connected to ground or V_{dd} , the value of v_{th} is a nonlinear function dependent primarily on $v_{sb} = v_s - v_{bulk}$. The rather significant variation of v_{th} with respect to v_{sb} leads us to comple-

ment our model with a nonlinear v_{th} model. Again, to expedite simulation, this nonlinearity is modeled in table.

The companion model in Figure 2 and the nonlinear v_{th} model are the primary d.c. components in our transistor model. To complete the model, we need also the capacitors, which are generally nonlinear. We chose to rely on a constant capacitance model for the gate, the overlap, the source/drain diffusion area, and the sidewall of the source/drain diffusion. It has been our experience that this simplistic model provides enough accuracy in real designs, where layout parasitics dominate and mask the nonlinearity in the capacitance. As a backup measure, nonlinear models are also available for the diffusions, and the gate. Gate capacitance can even be distributed over source and drain terminals depending on the operating region of the transistor [11, page 98].

To make the simulator practical and useful, we have also developed a utility program that will convert a variety of SPICE models into the table. This utility program works by generating test structures in SPICE format, running SPICE, and extracting the pertinent information from SPICE runs. By customizing the generation of SPICE netlist and the interpretation of SPICE results, user can effectively convert any SPICE models into our piecewise linear model, without requiring our utility program to understand the complicated analytical or empirical SPICE models at all.

3.2 Event driven approach

PowerMill's table lookup model offers a general, accurate, and efficient way to model a transistor. Model computation, however, is not the only cause for the slow run time and excessive memory usage in circuit simulators. For PowerMill to simulate designs with hundred of thousands of transistors in a couple of hours using an ordinary workstation, event-driven techniques [12, 10, 13], which decouple a large circuit into collections of smaller ones that can be handled separately, must also be used.

Unlike digital simulators and conventional timing simulators, in which events are defined as logic transitions, events in PowerMill are associated with a "significant" change in voltage level as in [14].

The processing of events is fairly conventional. Events are sorted using a wrapped-around array acting as an "event-wheel" [9]. By keeping track of and simulating the active portions of a circuit, the spatial and temporal latencies are naturally exploited.

The processing of an event involves the evaluation, or a transient simulation, of the channel connected component [9] in which the node resides.

3.3 Transient simulation

The evaluation of a channel connected component is accomplished using numerical integration techniques that are generally consistent with ordinary circuit simulators. In contrast to circuit simulation, however, we use the one-step-relaxation method (OSR) [15]. Sizes of time steps are carefully controlled to bound the amount of change in node voltages from one time point to the next. This bound ensures the validity of OSR, which is critical to maintain the accuracy of the result.

Since the solution method is an implicit one, such step size control is not always successful, in which case the solution is rejected and step size reduced, until the changes in voltages are small enough. Again, to ensure generality, nonlinear d.c. iterations can be requested by the user on selected parts.

Given a time step, capacitors are discretized. Transistor models are updated, i.e. the values of the current source, the conductance, and the transconductance in the companion model (Figure 2) are looked up again since the drain, source, and gate voltages have changed. A new nodal matrix is formed and solved with a sparse matrix solver.

Event propagation The transient simulation process is repeated and time advanced until

1. the circuit is settled; or
2. another input node to the channel-connected component being simulated will have an event; or
3. one of the nodes in the channel-connected component changes its voltage “significantly” if the node drives another channel-connected component.

In the last scenario, the node that will change “significantly” schedules an event corresponding to the time in the transient simulation. When the global simulation time advances to that time point, this event is processed as described in the previous section.

3.4 DC initialization

Before transient simulation starts, the circuit DC state must be solved. Instead of using nonlinear iteration which usually cannot converge for the size of circuits we want to handle, we take advantage of the fact that the circuit is mostly digital. By a topological sorting of the elements, followed by the propagation of input logic values through the elements in the topological order, we can properly initialize most, if not all, of the nodes. The nodes not initialized then undergo nonlinear iterations and will usually converge, since there are usually so few of them left. If there’re nodes that have trouble converging, we start the transient simulation anyway to allow them to settle, before any input vector comes in.

4 Example

To illustrate the accuracy of our approach, we show in Figure 3 a four-bit adder constructed from exclusive-OR gates and a pass transistor chain, also known as the Manchester carry chain. The presence of long pass transistor chains, precharged logic, and exclusive-OR gates, makes the accurate simulation very challenging.

In Figure 4 we show the voltage waveforms and current waveforms produced by PowerMill overlaid on those produced by SPICE. It is clear from the waveforms that PowerMill indeed has the requisite accuracy to deal with the nondigital behaviors of CMOS circuits.

<i>circuit type</i>	<i>number of transistors</i>	<i>number of vectors</i>	<i>CPU minutes</i>
32-bit adder	1,143	200	5
16-bit multiplier	9,476	20	5
controller	63,158	4,000	43
microprocessor	70,000	400	20
microprocessor	93,308	799	38
floating-point unit	350,000	3700	219
microprocessor	500,000	50	6 hours

Table 1: Run time summary of PowerMill. CPU times are collected on a Sun SPARC2 machine.

As mentioned earlier, PowerMill is intended for the current simulation of very large circuit blocks and entire chips. To show that it has the required efficiency, its run times for several large circuits are tabulated in Table 1.

5 Experience at ITT-Intermetall

The PowerMill simulator has been successfully used by ITT-Intermetall for a 0.8μ technology. PowerMill’s simulation result was found to match closely with measured results. For example a 61-stage ring oscillator was simulated with PowerMill to yield an average current of 0.176 mA, and a gate delay of 0.420 nanoseconds. This agrees very well with the measured results of 0.173 mA average current and 0.413 nanoseconds gate delay.

As another example, PowerMill was used to simulate a fast processor with 57,618 transistors operating at a 5 volt voltage supply and a 40 MHz clock. A total of 780 vectors was simulated in 99 minutes on a SUN SPARC1. The average current was found to be 24.9 mA, whereas measurements from an HP 82000 tester found the average current to be 23 mA. These two cases clearly provide ample evidence of the accuracy and speed of our approach.

6 Conclusion and on-going work

We have presented a set of unique and novel algorithms for the efficient and accurate simulation of current and power in large VLSI CMOS designs. Through comparisons with SPICE and benchmarks on large industrial examples, the accuracy, speed, and capacity of our approach are verified. To date, the program have been used extensively in over a 100 customer sites, and have indeed helped to uncover potential power problems.

As we discussed earlier, an accurate and fast power analysis tool such as PowerMill is only the first step toward a comprehensive set of CAD tools to help designers cope with power related problems. Issues such as power diagnostics, optimization, power bus design, and reliability assessment can all be addressed in a relatively straightforward manner based on such tools.

The speed and capacity of PowerMill, however, are still limited due to the broad scope of circuits the tool is designed to accommodate and the level of accuracy

it is to deliver. We continue to examine other approaches, such as the existing approaches discussed in an earlier section, which may provide greater speed and capacity with a narrower scope and somewhat lower accuracy.

References

[1] W. S. Song and L. A. Glasser. Power distribution techniques for VLSI circuits. *J. Solid State Circuits*, Feb. 1986.

[2] S. Chowdhury and M. A. Breuer. Minimal area sizing for power and ground nets for vlsi circuits. In *Proceedings of the 4th MIT Conference on Advanced Research in VLSI*, pages 141–169, 1986.

[3] J. R. Black. Electronmigration failure modes in aluminum metalization for semiconductor devices. *Proc. IEEE*, pages 1587–1594, Sept. 1969.

[4] R. Burch, F. Najm, P. Yang, and D. Hocevar. Pattern-independent current estimation. In *Proceedings of the Design Automation Conference*, 1988.

[5] F. Najm. Transition density, a stochastic measure of activity in digital circuits. In *Proceedings of the 28th Design Automation Conference*, 1991.

[6] M. Marculescu, D. Marculescu, and M. Pedram. Logic level power estimation considering spatiotemporal correlations. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, 1994.

[7] J. Monteiro, S. Devadas, B. Lin, C-Y Tsui, and M. Pedram. Exact and approximate methods of switching activity estimation in sequential logic circuits. In *Proceedings of the 1994 International workshop on low power design*, 1994.

[8] B. George *et al.* Power analysis for semi-custom design. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1994.

[9] Christopher Jay Terman. *Simulation tools for digital LSI design*. PhD thesis, MIT, Sept. 1983.

[10] B. R. Chawla, H. K. Gummel, and P. Kozah. MOTIS - an MOS timing simulator. *IEEE Trans. CAS*, Dec. 1975.

[11] Lance A. Glasser and Daniel W. Dobberpuhl. *The design and analysis of VLSI circuits*. Addison-Wesley, 1985.

[12] A. R. Newton. Techniques for the simulation of large-scale integrated circuits. *IEEE Trans. CAS*, Sept. 1979.

[13] S. P. Fan, M. Y. Hsueh, A. R. Newton, and D. O. Pederson. MOTIS-C: A new circuit simulator for MOS LSI circuits. In *Proceedings of ISCAS*, 1977.

[14] P. Odryna and S. Nassif. The ADEPT timing simulation program. *VLSI System Design*, March 1986.

[15] R. A. Saleh, J. E. Kleckner, and A. R. Newton. Iterated timing analysis and SPLICE1. In *ICCAD Digest of Technical Papers*, 1983.

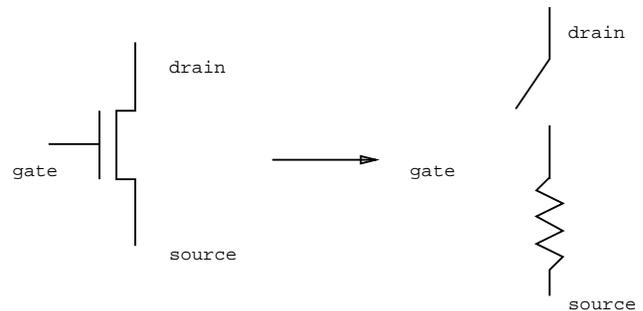


Figure 1: Transistor model in conventional switch-level timing simulators.

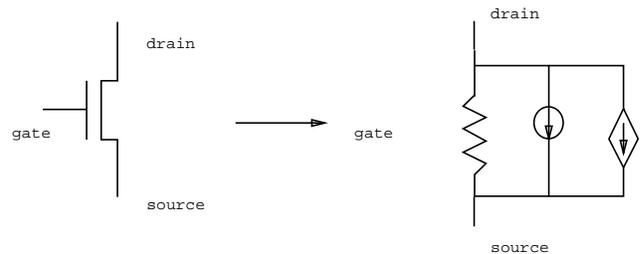
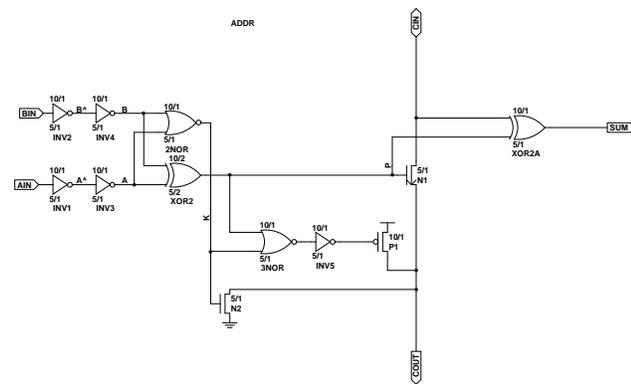


Figure 2: Transistor model in PowerMill and circuit simulators.



4B_ADDER

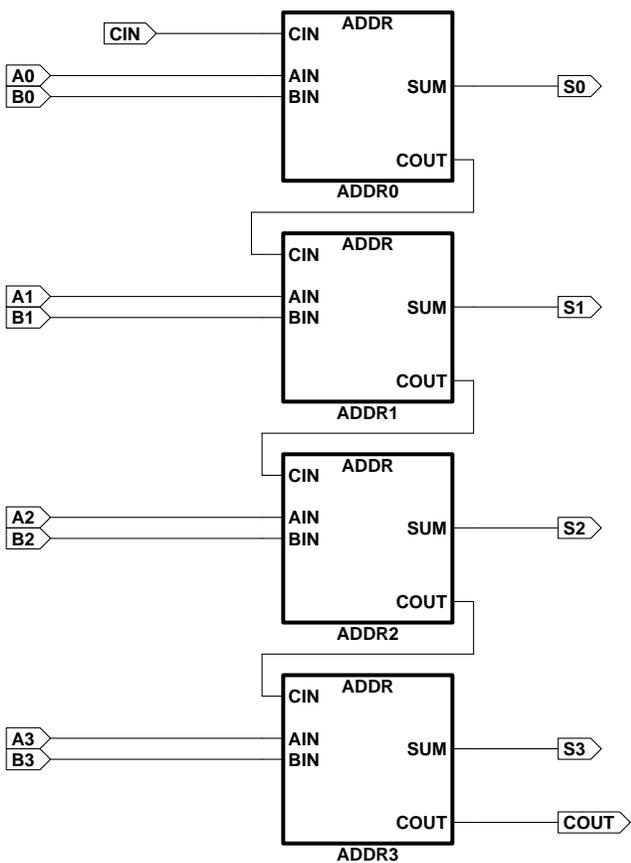


Figure 3: Top: a one-bit adder. Bottom: a four-bit Manchester carry adder constructed from the one-bit adder.

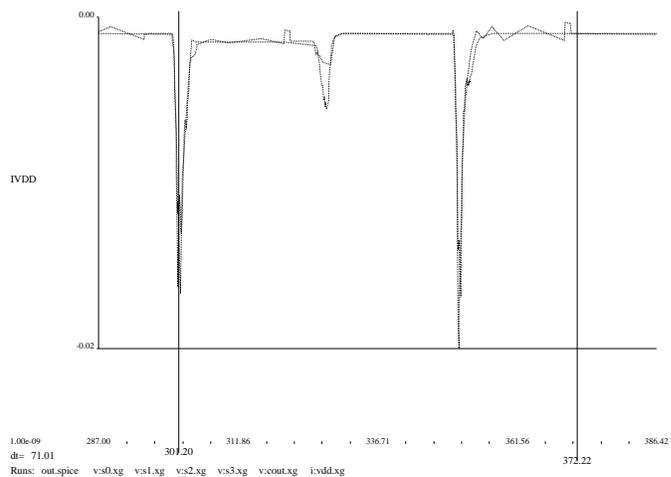


Figure 4: PowerMill's voltage and current waveforms overlaid on those of SPICE. The bottom plot is a more detailed view of a portion of the current waveforms.