Clock and Power Gating with Timing Closure

Arindam Mukherjee
University of North Carolina at Charlotte

Malgorzata Marek-Sadowska
University of California, Santa Barbara

Editor's note:
Assuming that delay is linearly dependent on local power supply voltage, the authors show how to set up an analysis to determine the effect of power supply variations on delay. This analysis can drive the introduction of clock gating, an increasingly popular technique for reducing dynamic power dissipation.

—Sani R. Nassif, IBM Austin Research Laboratory

HIGH PERFORMANCE in deep-submicron circuits incurs high power dissipation penalties. Power dissipation and currents drawn by circuits increase, as do the parasitics of power and ground (P/G) networks. Moreover, signal transition times decrease. These effects lead to increased noise in P/G networks. Because deep-submicron device delays are sensitive to such noise, timing violations can occur. Simulations show that timing violations in the presence of P/G noise can be as much as 30% of the clock period for a 0.25-micron pipelined processor core.

There are several runtime power reduction techniques that dynamically switch off a circuit's idle portions—for example, clock gating and dynamic clock management.[1,2] Clock gating disables the clock signal, thus idling circuit parts and saving dynamic power. Power gating disconnects idle subcircuits from the P/G networks, saving leakage power. Such savings are especially relevant in modern circuits with considerable amounts of on-chip memory. The use of this memory by active cells is a time-varying function of applications and design styles. A study of an integrated architectural- and circuit-level approach to reducing leakage energy dissipation in instruction caches demonstrated that power gating can reduce energy delay by up to 62%.[3]

When circuit portions are switched on and off, sharp current spikes and average current changes occur at the P/G nodes. These spikes and changes lead to time-varying *IR* drops and simultaneous-switching noise in the P/G networks, which in turn cause timing divergence. Device delay sensitivities to P/G noise increase with technology scaling because transistor cutoff voltage scales more slowly than supply voltage, so devices spend relatively more time in the cutoff and ohmic regions and less time in the saturation region.

Although inductive $L(dI/dt)$ switching noise is a major concern, techniques exist to reduce it, including incremental logic-block switching and decoupling-capacitor insertion. The other component of simultaneous-switching noise is *IR* drop, caused by resistive P/G wires. During clock or power gating, a subcircuit's current level changes from its previous value to a new, average ensemble value, at which it remains throughout the clock periods for which the gating is valid. During this time, the instantaneous current values exhibit high-frequency fluctuations on top of the ensemble value. If P/G networks have been designed to shield a subcircuit from $L(dI/dt)$ effects during switching caused by clock or power gating, the current fluctuations above the ensemble value are also filtered out because they are generally smaller than currents drawn during gate switching. In that case, voltage fluctuations due to switching decoupling capacitors, together with the *IR* drops caused by ensemble currents, become the main source of P/G noise. We refer to this noise as ensemble P/G noise.

Here, we introduce a timing-aware power reduction methodology that satisfies all timing constraints in the presence of ensemble P/G noise. (We assume that the P/G networks can shield $L(dI/dt)$ effects and that the only noise present is ensemble P/G noise.) Our experimental results have shown this methodology to be a promising solution to the problem of timing violations induced by clock and power gating in deep-submicron circuits.

0740-7475/03/$17.00 © 2003 IEEE Copublished by the IEEE CS and the IEEE CASS

Analysis models and problem formulation

Clock gate g_i in a subcircuit powered by a P/G node is a subset of the subcircuit's clock sinks, any of which can be disabled while the subcircuit executes operations. Power reduction occurs if the decrease in power dissipation caused by clock gating exceeds the increase caused by gating logic. The clock gate's frequency f_i is the relative occurrence of operations that trigger g_i, expressed as a percentage of the occurrence of all operations in the subcircuit powered by the P/G node.

We gate the clock signal to a synchronous element by using a logical AND between the clock signal and a gate control signal. A gated clock tree has masking AND gates at internal nodes, which the gate control signals selectively turn on and off to control the clock signal.[4] The masking gates load the clock trees similarly, whether or not clock gating is active. The clock period is determined by delays in the signal paths between latches, not by the load on a clock tree. Thus, clock gating incurs no delay penalty.

In the power-gating scheme we use, all blocks supplied from a single power node have their ground terminals connected to a virtual ground connection, g.[3] Several folded power-gating transistors of total gate width w_{cp} are connected between g and a ground node whose gate voltages are controlled by a charge pump. During the blocks' inactive phase, the power-gating transistors are turned off, so no current leakage path exists between power and ground meshes. Thus, blocks sharing the same virtual ground are simultaneously power gated. When power-gating transistors are present, output discharge delays of the active devices increase during the blocks' active phase. Simulations show that a path delay of up to 7% can occur in 0.25-micron technology if w_{cp} equals the total gate width of all transistors that discharge through power-gating transistors. This penalty is incurred naturally by the power-saving methodology, not our optimization, which only aims to minimize power under timing constraints.

Research has found that after the initial signal transitions caused by clock or power gating, a block's average power levels generally stabilize at some fixed value.[5] When we divide this power by the voltage across the block, we get the block's average current for the clock cycles in which the gating is valid. We similarly characterize all blocks for their average current values for all possible gatings. However, changes occur in average current levels over time, so ensemble P/G node voltages do not remain constant. Their fluctuations depend on charge redistributions among the decoupling capacitors.

Power reduction by clock or power gating is viable only if it remains valid for several subsequent instructions.

Our P/G networks are designed as meshes, with each branch modeled resistively and capacitively. We assume that one of the power nodes forming a square of the mesh supplies current to all active devices inside that square. The amount of current a node supplies equals the sum of the average currents drawn by all active devices inside its corresponding square. The decoupling capacitors attached to power nodes reduce simultaneous-switching noise during signal transitions. Experimenters have observed that for an operating frequency of around 250 MHz in 0.25-micron technology, RC modeling of the P/G meshes is sufficient, and inductive modeling is unnecessary.[6,7]

Here, we consider a placed, routed circuit with predesigned clock network and P/G meshes. We know the clock- and power-gating characteristics of the P/G nodes; the circuit maintains timing closure in the absence of any clock or power gating. We choose clock and power gates that maximize reduction in power dissipation while ensuring that no timing violations occur in the presence of ensemble P/G noise.

Our target applications include but are not limited to processor core circuits. Once we have found the compatible clock and power gates for different instruction sequences, we can program them into micromemory. Power reduction by clock or power gating is viable only if it remains valid for several subsequent instructions, because circuits waste transient energy while entering and exiting any gated state.

Timing-related potential constraints

Researchers have proved analytically that in deep-submicron circuits, delays of active elements change linearly with P/G noise.[3] Closed-form expressions, relating a buffer's change of delay ($\Delta delay$) with the change in potentials of its P/G nodes, are shown as

$$\Delta delay = f_1(U - u) + f_2(v - V) \quad (1)$$

where f_1 and f_2 are coefficients dependent on input transition time, gate load, and device or technology para-

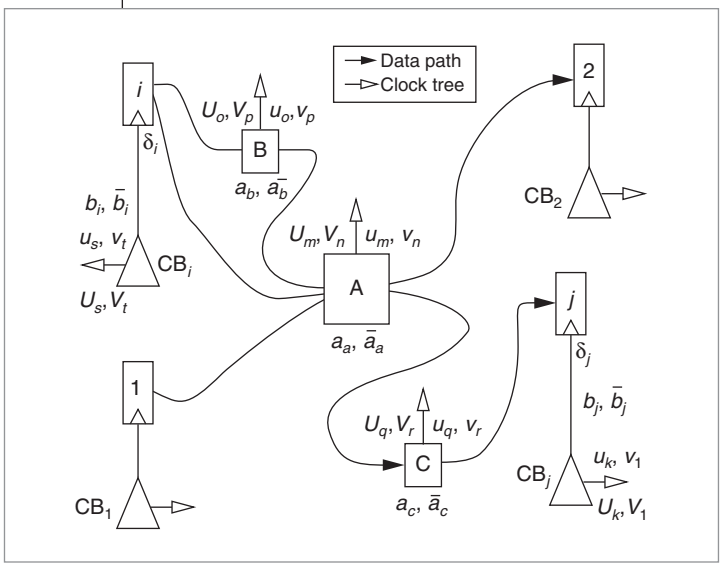

Figure 1. A placed and routed sequential netlist.

meters. U and V are the initial average power and ground potentials; u and v are the buffer's changed average P/G node potentials caused by P/G noise, which in our case is induced by clock and power gating. We observed experimentally that the relation expressed by Equation 1 also holds for more complex gates within certain ranges of P/G noise.

Figure 1 shows a placed and routed netlist of latches and logic elements, with each logic block or buffer characterized by six parameters. For instance, U_m and V_n represent block A's initial power and ground node potentials (without clock and power gating), whereas u_m and v_n represent changed supply node potentials caused by gating. Parameters a_a and $\bar{a}_a$ denote the sensitivities of block A's delay to power and ground potential changes. These parameters correspond to the f_1 and f_2 coefficients in Equation 1. Latch i has an associated clock buffer denoted by CB_i. Such clock buffers are characterized similarly to logic blocks.

In Figure 1, parameters b_i and $\bar{b}_i$ represent the delay sensitivities of CB_i to P/G noise. A single clock buffer can clock many latches, and our clock tree synthesizer makes a zero-skew clock subtree rooted in the clock buffer. This subtree has no active buffers. Thus, P/G noise does not affect the subtree's zero-skew nature. For latches clocked by different buffers, the clock signal arrival times account for the skew between their clock signals. The clock signal arrival time to latch i, with respect to the clock source, is denoted δ_i. Simply put, the latches are assumed to have no delays and thus are unaffected by P/G noise. By considering latched synchronous circuits, we can accommodate multicycle paths.

Each latch, combinational block, or interconnect buffer can be perceived as a vertex in a graph representation. A latch is an l-vertex, and a block or buffer is a c-vertex. A c-vertex, x, is weighted by the internal delay (d_x) of the corresponding block or buffer. A directed edge, e_{xy}, exists between a pair of vertices (x, y), from x to y, if the output of the element represented by x is connected to the input of the element represented by y. All vertices connected by incoming edges to vertex k form a fan-in of k, $FI(k)$. Similarly, all vertices connected by outgoing edges from vertex k form fan-out $FO(k)$.

We denote the circuit's clock period by ϕ. We represent the required time of a data signal at the input of vertex z by τ_z and the arrival time at the same input by t_z. If z is a c-vertex, its internal delay d_z has a nonzero value; otherwise, it is 0. If z is an l-vertex, the clock arrival time to z is δ_z.

Consider a path terminating at l-vertex z. For timing correctness, data caused by any previous input vector should be latched into z when data caused by the current vector is present in the input paths to z. In this case, τ_z is one clock period after the previous clock tick but at or before δ_z, minus latch setup time T_l. Thus,

$$\tau_z = \phi + \delta_z - T_l$$

For a path starting at l-vertex z, t_z equals δ_z. For c-vertex z, τ_z equals the minimum among all the required times of the vertices in $FO(z)$, minus d_z. Thus,

$$\tau_z = \min\left(\forall_{y \in FO(z)} \tau_y - d_z\right)$$

Similarly, if $y \in FO(z)$ is a c-vertex, we use the required times of all vertices in the fan-out set of y to find τ_y. This recursive computation continues until we reach all the l-vertices that terminate paths. Thus, we calculate the required time at z through backward propagation of the required times at the output latches.

The arrival time at the input of c-vertex z is the maximum among the summations of arrival time t_x at any vertex x in $FI(z)$ and internal delay d_x of x. Thus,

$$t_z = \max\left[\forall_{x \in FI(z)}(t_x + d_x)\right]$$

Similarly, if $x \in FI(z)$ is a c-vertex, we use the arrival times of the vertices in the fan-in set of x to find t_x. This recursive computation continues until we reach the l-

vertices at starting points. Thus, we calculate the arrival time at z through forward propagation of the arrival times at the input latches.

We can calculate all logic element delays, as well as clock signal arrival times, for their initial respective supply node potentials in the absence of any clock and power gating.

For timing closure, τ_z must be greater than or equal to t_z for any c-vertex z. Using the expressions described, we get

$$\max\left[\forall_{x\in FI(z)}(t_x + d_x)\right] \le \min\left[\forall_{y\in FO(z)}(\tau_y - d_z)\right] \quad (2)$$

The left-hand side of Equation 2 gives the latest arrival time of the signals to z, which corresponds to the longest path delay in the fan-in cone of z. This is the long-path constraint. Also, the earliest-arriving data for the current vector should not reach z before the clock signal to z arrives and latches to the data for the previous vector. This is the short-path constraint.

Let λ_z represent the required time of a data signal at the input of vertex z. The required time at the input of c-vertex z should be less than or equal to its arrival time:

$$\min\left[\forall_{y\in FO(z)}(\lambda_y - d_z)\right] \le \max\left[\forall_{x\in FI(z)}(t_x + d_x)\right] \quad (3)$$

In the presence of P/G noise, delays of all active devices, including those of the clock buffers, change according to Equation 1. We represent the change of arrival time of the clock signal to any latch i from its corresponding clock buffer by Equation 4a and the change in delay of any active device or block by Equation 4b:

$$\Delta\delta_i = \sum_{n\in N} b_j\left(U_n - u_n\right) + \sum_{m\in M} \bar{b}_j\left(v_m - V_m\right) \quad (4a)$$

$$\Delta d_k = a_k\left(U_n - u_n\right) + \bar{a}_k\left(v_m - V_m\right) \quad (4b)$$

where the elements in sets N and M are the power and ground nodes supplying the buffers on the path in the clock from its root to CB_i, and j is any such clock buffer.

The inequalities in Equations 2 and 3 must be satisfied in the presence of clock- and power-gating-induced P/G noise. These expressions capture the effect of P/G noise on device delays in the changed values of the d_x and d_z terms. Changes in clock signal arrival times affect the t_x and τ_y terms. Thus, we can express inequalities 2 and 3 in terms of P/G noise, and we can capture timing constraints as potential constraints.

Power and ground noise analysis

Here, we consider P/G meshes with multiple external pads, designed to satisfy electromigration, minimum-width, and maximum-voltage-drop constraints. The circuit timing is correct in the absence of gating. During clock and power gating at a P/G node, the average current drawn from the node changes. This leads to potential change in the supply node (P/G noise) and hence to delay changes in the circuit, according to Equation 1.

Two P/G nodes are related if they supply different active elements and at least one path exists that includes both active elements. A power node and a ground node are related if they supply the same active device.

When a certain gating replaces another at P/G node i, the change in average current ΔI_i causes a change in node charge by $\Delta q_i = \Delta I_i/2\phi$, where ϕ is the clock period. The effective decoupling capacitor[7] at node i is denoted dc_i. The maximum voltage fluctuation (Δw_i) above the steady-state value at the node is determined by $\Delta w_i = \Delta q_i/dc_i$. This is the measure of simultaneous-switching noise in our P/G mesh model. We estimate an upper bound of the *IR* drop at a P/G node in the steady state by using Dijkstra's all-pair-shortest-path algorithm[8] to find the least-resistive paths from a single external supply pad to all the grid points. The total resistance of such a path to grid point i is R_{ii}, the upper bound of current-path resistance. For a circuit with multiple external P/G pads, we assume that a P/G node is supplied by its closest P/G pad.[7] A direct consequence of this assumption is a slight overestimate of the current drawn from the P/G pad and the P/G noise at the node. Simulations have shown errors of less than 10%.[7]

The initial potential of power mesh node i is V_i. If a clock or power gate is selected such that ΔI_i is the decrease in the current drawn from i, the upper bound of P/G noise is

$$\Delta v_{ij} = R_{ii} \times \Delta I_i + \Delta w_i$$

The change in current at node i also affects the potentials of nodes in its vicinity because of the sharing of P/G mesh segments between the nodes. For any such node j, the upper bound of P/G noise in node i is

$$\Delta v_{ij} = R_{ij} \times \Delta I_i + \Delta w_{ij}$$

where R_{ij} is the upper bound on the common resistance between nodes i and j, and is given by $R_{ij} = \min(R_{ii}, R_{jj})$. The maximum change in voltage at j caused by gating at i is denoted Δw_{ij}, which we estimate from Δw_i using

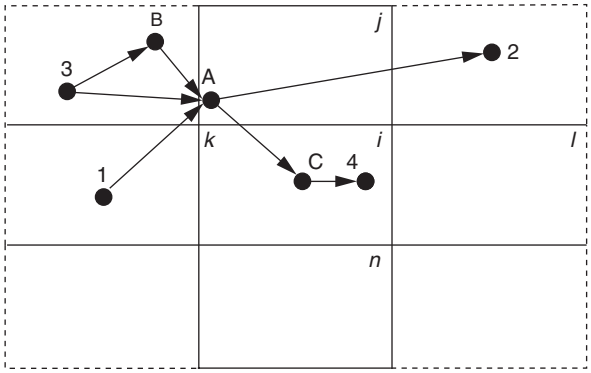

Figure 2. P/G node vicinity in a power mesh, with a superimposed subcircuit graph. Solid lines outline T_i, the immediate vicinity of *i*.

the spatial locality of *i* and *j*. We analyze the ground mesh similarly.

Figure 2 shows a P/G node vicinity in a power mesh. Considering power node *i*, suppose the current drawn from *i* has changed by ΔI_i. For efficient analysis, we assume that P/G noise induced by this change is limited to grid nodes {*i*, *j*, *k*, *l*, *n*} in the immediate vicinity of *i*. In Figure 2, solid lines delineate this vicinity, T_i. A subcircuit graph is superimposed on the power mesh. All vertices with an incoming edge from a vertex in T_i to a vertex not in T_i form elements of output set *O*. Similarly, all vertices not in T_i with an outgoing edge to a vertex in T_i form elements of input set *P*. In this case, $O = \{2\}$, and $P = \{3, B\}$. The lowercase *s* denotes a path that starts from either a vertex in *P* or an l-vertex in T_i and ends at either a vertex in *O* or an l-vertex in T_i. The uppercase *S* represents the set of all such paths. Analysis for the ground mesh is similar.

In the presence of power noise, a path *s* from *x* to *y* fulfills the condition of Equation 2:

$$\sum_{f \in s - \{x, y\}} \left(d_f + \Delta d_f\right) + \left(t_x + \Delta t_x\right) - \left(\tau_y + \Delta \tau_y\right) \leq 0$$

Rearranging this equation gives

$$\sum_{f \in s - \{x, y\}} \Delta d_f + \Delta t_x - \Delta \tau_y \leq const_1$$

where $const_1$ is a positive quantity given by

$$\tau_y - t_x - \sum_{f \in s - \{x, y\}}$$

If *x* belongs to *P*, Δt_x is 0, and *y* belongs to *O*, then Δt_y is 0. Equation 3 satisfies the short-path constraint for the same path when

$$\sum_{f \in s - \{x, y\}} \Delta d_f + \Delta t_x - \Delta t_y \geq const_2$$

where $const_2$ is a negative quantity given by

$$\lambda_y - t_x - \sum_{f \in s - \{x, y\}} d_f$$

The clock arrival time of any l-vertex *f* supplied by a power node *m* in T_i changes according to Equation 4a. This delay change is $b_f(U_m - u_m)$, which is nonzero only if a clock buffer for the l- vertex exists in T_i. The delay of c-vertex *f* powered by *m* in T_i changes by $a_f(U_m - u_m)$ according to Equation 4b. Thus, timing relations in the presence of P/G noise can be expressed

$$const_2 \leq \sum_{f \in s,\ m \in M} \left[c_f \left(U_m - u_m\right)\right] \leq const_1$$

where c_f is either a_f or b_f. Power node *m* can supply several vertices in a subset of vertices, *E* on path *s*.·In this expression, we can add the delay sensitivity coefficients of all vertices in *E* into the parameter

$$h_m = \sum_{e \in E} c_e$$

Thus, the inequality becomes

$$const_2 \leq \sum_{m \in M} \left[h_m \left(U_m - u_m\right)\right] \leq const_1 \qquad (5)$$

We will use this expression to find the effect of the potential variation at P/G node *i* on the range of potential variation of another node in its vicinity T_i. For any P/G node *j* in T_i, we set the P/G noise of other nodes in T_i to 0. Thus, Equation 5 becomes $const_2 \leq h_j(U_j - u_j) \leq const_1$ or

$$U_j - const_1/h_j \leq u_j \leq U_j + |const_2|/h_j \qquad (6a)$$

if h_j is positive, and

$$U_j - |const_2|/|h_j| \leq u_j \leq U_j + const_1/|h_j| \qquad (6b)$$

if h_j is negative.

If h_j is positive, we use expression 6a and get

$$E_j(s) = U_j - const_1/h_j$$

$$H_j(s) = U_j + |const_2|/h_j$$

If h_j is negative, we use expression 6b and get

$$E_j(s) = U_j - |const_2|/|h_j|$$

$$H_j(s) = U_j + const_1/|h_j|$$

If h_j is 0, it means that the active devices' delays do not change with P/G noise at node j, and hence there is no limit to potential fluctuations at m.

The sensitivity of upper or lower bounds of j to the noise at i is $\eta_{ij}(s) = h_i/h_j$. The potential bounds and sensitivities of the nodes in T_i depend on path s.

The potential of node j in the presence of clock and power gating is v_j. The difference between its initial and final values arises from two factors: change in potential of j caused by clock and power gating associated with its mesh square, and change in potential at other nodes. Final node voltage v_j is determined by the superposition of potential changes caused by these factors. The upper and lower bounds of v_j depend on potential changes at related nodes. Let i and j be two related nodes supplying some active devices whose corresponding vertices can exist in a set of paths $S_1 \subseteq S$. Timing closure is maintained in the presence of clock and power gating if the following relations hold between i and j for every such path s:

$$E_j(s) + \eta_{ij}(s)\Delta v_{ii} \le v_j \le H_j(s) + \eta_{ij}(s)\Delta v_{ii} \quad (7)$$

Under different P/G node voltages, the actual upper bound of v_j is

$$\min\,[H_j(s) + \eta_{ij}(s)\Delta v_{ii}],\ \forall s \in S_1$$

where i is a related node of j, and $s \in S_1$. Similarly, the actual lower bound of j under the different node voltages is

$$\max\,[E_j(s) + \eta_{ij}(s)\Delta v_{ii}],\ \forall s \in S_1$$

For power (ground) node z with potential $u_z(v_z)$, the upper potential slack (S_z) is the difference between the actual upper bound of $u_z(v_z)$ and $u_z(v_z)$. The lower potential slack ($\overline{S}_z$) is the difference between $u_z(v_z)$ and the actual lower bound of $u_z(v_z)$. The actual value of v_j depends on the switching not only at node i but also at other nodes in its vicinity.

On average, our method saved 91% of the power saved by a method without timing considerations and resulted in no timing violations.

Selecting clock and power gatings

For a placed and routed circuit with predesigned P/G and clock networks, we selectively use clock and power gating to maximize power savings, subject to the satisfiability of timing constraints (Equations 2 and 3) in the presence of P/G-noise-induced delay changes. Our tool, Catar, implements a modified gate-sizing algorithm.[9] Gate sizing optimizes for power by replacing faster gates of large drive strengths with slower ones of smaller strengths. During this process, we must satisfy all timing constraints. Similarly, we aim to maximize the power reduction achieved by clock and power gating in a way that satisfies timing relations captured as potential relations in Equation 7.

Experiments and results

We have implemented our timing-aware power reduction algorithm as a C++ program running on the Sun Solaris platform. We used a pipelined AM2901 processor core for our experiments. Using a commercial tool flow, we synthesized a VHDL model of the processor and laid it out in 0.25-micron technology, together with a gated clock tree and gated P/G meshes. The circuit had about 35,000 gates, with an average of 1.3 nets per gate. The total number of external pins was 42. The P/G networks were synthesized as 70×70 meshes.

We found the actual values of resistances and capacitances of P/G wire segments through extraction and back-annotation of the layouts, using a commercial tool. We then performed timing analysis with the extracted parasitics. We chose a 4-ns clock period so that there were no timing violations in the absence of clock and power gating. We also synthesized the programmable logic required to control the gates of the power-gating transistors. The total increase in area from

Table 1. Comparison of simulation results from proposed method and greedy_nt method.

Application	Proposed method's power savings (% of greedy_nt total)	No. of violations using greedy_nt method
Arithmetic intensive	86	30
Logic intensive	92	34
Register intensive	95	27
Memory intensive	90	28
Average	91	30

added logic was less than 3%, contributing about 2% to 3% of the total average dissipation of circuit power.

Given the processor's instruction set architecture, we used four different instruction sequences for benchmarking, corresponding to different application programs. Each AP required a specific set of clock and power gates. The maximum power savings achievable through gating varied from 30% to 45% of the total circuit power for the different applications. Table 1 compares our optimization method with a heuristic that we developed, called greedy_nt. The greedy_nt method applies clock and power gating to maximize power savings in a greedy manner; it does not account for timing violations these gatings can cause. The first column of the table lists different APs, the second compares power savings of our method with those of greedy_nt, and the third shows the number of paths with timing violations that occurred with greedy_nt. This data came from Spice simulations of the back-annotated P/G layouts for clock and power gates chosen by the greedy_nt method.

We observed that with greedy_nt, the maximum timing violation was as much as 30% of the clock period. Two concentrations of nets had timing violations—one around 5 ns and the other around 1 ns, corresponding to 12% and 25% of the clock period. The logical and physical regularity of the processor's computational parts might have contributed to this result. Comparable simulations for the clock and power gates chosen by our method resulted in no timing violations.

On average, our method saved 91% of the power saved by the greedy_nt method, which also had an average of 30 timing violations. After running simulations for P/G node vicinities of different sizes, we noticed that on average our method predicts 1.2 to 2 times the actual number of violations as the size of the vicinities decreases, whereas the optimization runtimes scale up linearly according to the sizes of the vicinities. The third column of Table 1 reflects vicinities that include all P/G nodes separated from a particular one by a single mesh segment. This gives us reasonable accuracy for runtimes.

A typical optimization's CPU time was less than 5 minutes. The complexity of our algorithm is linear in the number of P/G nodes, whereas the complexity of the constraints is linear in the number of nodes in the circuit graph. This shows an advantage of a node-based algorithm over path-based ones.

HERE, WE HAVE USED average current values to analyze the ensemble behavior of P/G noise and its effect on gate delays. However, the effect of inductance on P/G noise is substantial during switching from one gating state to the other, and we plan to incorporate this effect in our future work. Also, because deep-submicron gate delays depend on input vectors and slew rates, a gate has a timing window for switching, as opposed to a single delay value. We are exploring the integration of timing window analysis in our optimization method. ■

References

1. Q. Wu, M. Pedram, and X. Wu, "Clock-Gating and Its Application to Low Power Design of Sequential Circuits," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 3, Mar. 2000, pp. 415-420.
2. D. Brooks, and M. Martonosi, "Value-Based Clock Gating and Operation Packing: Dynamic Strategies for Improving Processor Power and Performance," *ACM Trans. Computer Systems*, vol. 18, no. 2, May 2000, pp. 89-126.
3. M. Powell et al., "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories," *Proc. ACM Int'l Symp. Low-Power Electronics and Design*, ACM Press, 2000, pp. 90-95.
4. J. Oh and M. Pedram, "Gated Clock Routing Minimizing the Switched Capacitance," *Proc. Design, Automation and Test in Europe* (DATE 98), IEEE CS Press, 1998, pp. 692-697.
5. L. Smith et al., "Power Distribution System Design Methodology and Capacitor Selection for Modern CMOS Technology," *IEEE Trans. Advanced Packaging*, vol. 22, no. 3, Aug. 1999, pp. 284-291.
6. L.-H. Chen, M. Marek-Sadowska, and F. Brewer, "Coping with Buffer Delay Change due to Power and Ground Noise," *Proc. 39th Design Automation Conf.* (DAC 02), ACM Press, 2002, pp. 860-865.
7. S. Zhao, K. Roy, and C.-K. Koh, "Decoupling Capacitance Allocation and Its Application to Power

Supply Noise-Aware Floorplanning," *IEEE Trans. Computer-Aided Design of Circuits and Systems*, vol. 21, no. 1, Jan. 2002, pp. 81-92.
8. P.K. Chan and M.D.F. Schlag, "Bounds on Signal Delay in RC Mesh Networks," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 6, June 1989, pp. 581-589.
9. O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization," *IEEE Trans. VLSI Systems*, vol. 5, no. 4, Dec. 1997, pp. 465-472.

Arindam Mukherjee is a faculty member in the Department of Electrical and Computer Engineering of the University of North Carolina at Charlotte. His research interests include electronic design automation of power and ground network design and analysis, power management and low-power circuits, designing for predictability in deep-submicron circuits, clock network synthesis, floorplanning, placement and routing, IP core integration, and programmable fabrics. Mukherjee has a BE in electronics and telecommunications engineering from Jadavpur University, India; and an MS and a PhD, both in electrical and computer engineering, from the University of California, Santa Barbara. He is a member of the IEEE.

Malgorzata Marek-Sadowska is a professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. Her research interests include physical design automation, logic synthesis, and VLSI testing. Marek-Sadowska has an MS in applied mathematics and a PhD in electrical engineering, both from the Technical University of Warsaw, Poland. She is a member of the IEEE.

■ Direct questions and comments about this article to Arindam Mukherjee, Dept. of Electrical and Computer Eng., University of North Carolina, Charlotte, NC 28223; amukherj@uncc.edu.