An Effective Power Mode Transition Technique in MTCMOS Circuits

Afshin Abdollahi  
University of Southern California  
afs@usc.edu

Farzan Fallah  
Fujitsu Labs. of America  
farzan@us.fujitsu.com

Massoud Pedram  
University of Southern California  
pedram@ceng.usc.edu

Abstract - The large magnitude of supply/ground bounces, which arise from power mode transitions in power gating structures, may cause spurious transitions in a circuit. This can result in wrong values being latched in the circuit registers. We propose a design methodology for limiting the maximum value of the supply/ground currents to a user-specified threshold level while minimizing the wake up (sleep to active mode transition) time. In addition to controlling the sudden discharge of the accumulated charge in the intermediate nodes of the circuit through the sleep transistors during the wake up transition, we can eliminate short circuit current and spurious switching activity during this time. This is in turn achieved by reducing the amount of charge that must be removed from the intermediate nodes of the circuit and by turning on different parts of the circuit in a way that causes a uniform distribution of current over the wake up time. Simulation results show that, compared to existing wakeup scheduling methods, the proposed techniques result in a one to two orders of magnitude improvement in the product of the maximum ground current and the wake up time.

Categories & Subject Descriptors
B.6.3 [Hardware] Logic Design: Design aids – Automatic synthesis, Optimization.

General Terms

1. Introduction
Multi-threshold CMOS (MTCMOS) technology provides low leakage and high performance operation by utilizing high speed, low Vt transistors for logic cells and low leakage, high Vt devices as sleep transistors. Sleep transistors disconnect logic cells from the power supply and/or ground to reduce the leakage in sleep mode. In this technology, also called power gating, wake up latency and power plane integrity are key concerns. Assuming a sleep/wake up signal provided from a power management unit, an important issue is to minimize the time required to turn on the circuit upon receiving the wake up signal since the length of wake up time can affect the overall performance of the VLSI circuit. Furthermore, the large current flowing to ground when sleep transistors are turned on can become a major source of noise on the power distribution network, which can in turn adversely impact the performance and/or functionality of the other parts of the circuit. There is trade off between the amount of current flowing to ground and the transition time from the sleep mode to the active mode.

In this paper we introduce an approach for reducing the transition time from sleep mode to active mode for a circuit part while assuring power integrity for the rest of the system by restricting the current that flows to ground during the transition. The problem is to minimize the wakeup time while constraining the current flowing to ground during the sleep to active mode transition. Our approach is comprised of the following steps. First the discharge patterns of all logic cells are obtained. Next all cells in the circuit are clustered to a minimum number of clusters in such a way that the total discharge current of each cluster does not exceed a given threshold. Another constraint is imposed on clustering which will prevent flowing of short circuit current during wakeup time. This constraint is handled by introducing a constraint graph and not allowing two cells with an edge between them in the same cluster. Finally for each cluster a single sleep transistor is assigned which is associated with a limited performance penalty and for each sleep transistor, a sleep/wake up signal is assigned. The wakeup times for the clusters are optimized to achieve minimum wake up time while maintaining a given threshold on overall discharge current.

Section 2 describes the previous work. In Section 3 we present the key observations that our technique is based on. Section 4 presents problem statement and our method for solving it, while an alternative method for reducing the ground bounce is proposed in Section 5. Simulation results are presented in Section 6. Section 7 concludes the paper by briefly describing our future plan.

2. Previous Work
Optimal sizing of the sleep transistors for an arbitrary circuit to meet a performance constraint is an important design problem. Sleep transistors cause logic cells to slow down because of the voltage drop across the functionally-redundant sleep transistors and due to the increase in the threshold voltages of logic cell transistors as a result of the body effect. The performance penalty of a sleep transistor depends on its size and the amount of current that goes through it. In [1], sleep transistors are modeled as resistors and subsequently sized according to the following approximation for propagation delay: $T_{pad} \approx C_i V_{dd}/(V_{dd} - V_t - V_i)^\alpha$ where $C_i$ is the total load capacitance, $V_{dd}$ is the supply voltage, $V_t$ is the voltage drop across the sleep transistor, and $V_i$ is the threshold voltage and $\alpha$ is a constant modeling the short channel effects. This delay model is used to bound the performance penalty for the worst case input vector. In [2], the authors propose a different method for sizing the sleep transistors. They first size the sleep transistor of each cell to limit the performance degradation to a specified level. Next, they merge sleep transistors whose discharge current patterns are mutually exclusive based on a unit delay model. In [3], the authors use a more precise delay model to do the same steps. In [4], the authors propose a power gating structure to support an intermediate power-saving mode and a traditional power cut-off mode. The idea is to add a PMOS transistor in parallel with each NMOS sleep transistor whereby applying zero voltage to the gate of the PMOS transistor the circuit can be put in the intermediate mode. In the intermediate mode leakage reduction and data retention are realized. Furthermore, the magnitude of power supply voltage fluctuations during power-mode transitions is reduced by
transitioning through this intermediate mode while changing between sleep and active modes. In the cut-off mode the gate of the additional PMOS transistor is connected to \( V_{dd} \).

None of these works attempt to minimize the wake up time and the noise generated by the power gating structure and until recently only few researchers have addressed this problem. In [5] the authors introduce two power mode transition techniques to reduce the ground bounce while turning on the circuit. Instead of quickly turning on a large sleep transistor to suddenly reduce the resistance between the virtual ground and the (actual) ground, they propose to gradually reduce the resistance of the sleep transistor in order to limit the peak current flowing to the ground. This can be accomplished by employing one of the following two methods:

**Parallel Sleep Transistors (Parallel-ST):** Use of parallel-connected sleep transistors with gradually increasing widths (cf. Figure 1.)

The sleep transistors are turned on in several time steps, starting from the smallest one. Since the voltage of the virtual ground is initially at its maximum value, a relatively high resistance value is used to discharge it; this limits the peak current. In the subsequent time steps, the resistance of the path between virtual and actual grounds is reduced by turning on wider sleep transistors.

**Figure 1. Power gating structure consisting parallel sleep transistors.**

**Staircase Sleep Signal (Staircase-SS):** Use of a single sleep transistor, but turning it on gradually. Initially a voltage less than \( V_{dd} \) is used to weakly turn on the sleep transistor and thus, somewhat reduce the voltage of the virtual ground. In subsequent steps, the sleep transistor is turned on more strongly to further reduce the resistance between the virtual and actual grounds.

The two methods of [5] are restricted to using one sleep signal for the entire circuit block and provide only a temporal solution to the peak current flow problem. In contrast, in this paper, we provide an efficient *spatio-temporal solution* with its supporting power gating structure (i.e., with the ability to turn on different logic cells in the circuit block at different times.) This solution enables us to minimize the wake up time subject to an upper bound constraint on the total maximum current through the sleep transistors.

### 3. Key Observations

It is a well known fact that there is no need to have both NMOS and PMOS sleep transistors to encapsulate a logic cell. In particular, NMOS sleep transistors can be used to separate the (actual) ground from the virtual ground of the logic cell. Upon entering the sleep mode, a circuit block is disconnected from the ground. This causes the voltage levels of some intermediate nodes in the circuit block to rise toward \( V_{dd} \). When the circuit block is woken up, the nodes will transition to zero. This transition in turn causes the logic cells in the immediate fanout of the node to carry a potentially large amount of short-circuit current as explained next. Consider the inverter chain shown in Figure 2, which is connected to the ground through an NMOS sleep transistor.

**Figure 2. A chain of four inverters with an NMOS sleep transistor.**

If the input of the circuit is low, then, in the active mode (i.e., \( SLEEP=0 \)), \( V_{A}=V_{C}=V_{E}=V_{G}=0 \) and \( V_{B}=V_{D}=V_{DD} \). When entering the sleep mode, the voltages of B and D do not change, but the voltages of C, E, and G gradually increase and will be equal to \( V_{DD} \) if the sleep period is long enough (note the driver of signal A is not controlled by the SLEEP signal). This happens because the leakage through the PMOS transistors will charge up all the floating capacitances. Figure 3 shows the voltage waveforms of nodes C, E, and G generated by HSPICE simulation. While turning on the sleep transistor, nodes G, C and E discharge as depicted in Figure 4.

**Figure 3. Voltage waveforms for nodes C, E and G of the circuit in Figure 2 when the circuit is in the sleep mode.**

**Figure 4. Voltage waveforms for nodes C, E and G of the circuit in Figure 2 when the circuit is transitioning from the sleep to active mode.**

As one can see when the voltage of G quickly reaches its final value, the voltages of C and E are still between zero and \( V_{DD} \). This results in a significant amount of short circuit current in the logic cells driven by nodes C and E since these nodes turn on both transistors of the inverters present in their fanout.

**Figure 5. Total current flowing to ground while turning on the circuit.**

The current shown in Figure 5 flowing through the sleep transistor is the result of not only discharging the accumulated charge in some intermediate nodes (i.e., C, E, and G in the inverter chain example), but also the short circuit current flowing through some logic cells of the circuit (e.g., the third inverter in the chain which
is driven by signal C). The smaller the number of nodes that are discharged, the smaller the amount of current that flows to ground.

Our design of a power gating structure and a wake up strategy are driven by the desire to avoid short circuit currents and spurious transitions by appropriately clustering the sleep transistors and by turning them on at proper times. The basic idea is to turn on each cell only if the voltage levels of the logic cells in its fanin have already reached their final values.

Consider an inverter chain with one sleep transistor per cell as depicted in Figure 6.

![Figure 6. A chain of inverters with separate sleep transistors.](image)

If we turn on the sleep transistors one at a time starting from the first inverter on the left, the short circuit current will be zero. The reason is that when each cell is turned on, its fanout cell continues to stay in the sleep mode. Therefore, the possible transition of the output node of the logic cell does not result in any short circuit current in its fanout cell. Furthermore, there will be no spurious transition in the circuit since the inputs of the logic cells that have been turned on will not change at a later time. Figure 7 shows the total current flowing to ground while turning on the circuit of Figure 6 by employing this wake up strategy. As we can see, compared to the data of Figure 5, the maximum current in Figure 7 is reduced from 375µA to 280µA.

![Figure 7. Total current flowing to ground while turning on the circuit.](image)

Notice that there are several peaks in the current waveform of Figure 7. This is due to the fact that the sleep transistors are turned on in four steps. This obviously increases the wake up time. Note that in Figure 6, we can simultaneously turn on the first and third inverters before turning on the second and forth inverters without producing any short circuit current. This will reduce the wake up delay of the circuit. The next section proposes an algorithm for applying this idea to a general combinational circuit.

## 4. Wakeup Signal Scheduling

Let $I_{\text{TURNON}}$ be the waveform over time of the sum of currents flowing to ground during sleep to active mode transition. In this paper the objective is to cluster logic cells to a minimum number of clusters and also devise a wake up (turn-on) strategy (i.e., a scheduling of the sleep/wakeup signals to activate clusters of logic cells in the circuit) which minimizes the turn-on time ($T_{\text{TURNON}}$) subject to a constraint on the maximum total current flowing through the sleep transistors (i.e., $\text{MAX}(I_{\text{TURNON}}) < I_{\text{MAX}}$ where the maximum is taken over all time instances.) Notice that after the clustering phase, a sleep transistor and a corresponding sleep/wakeup signal is assigned to each cluster. The size of sleep transistors can be determined using well known methods some of them presented in [2] and [3]. It is assumed that the circuit has been in the sleep mode for a sufficiently long period of time (about 100µs as can be seen in Figures 3) so the output voltages of all its logic cells have raised to their final steady state level. Also the vector that is applied to the circuit’s primary inputs upon entering and during the sleep mode is known and remains fixed during the wake up time. We call this input vector the sleep vector. With these assumptions the precise problem statement is as follows:

**Wakeup Signal Scheduling (WSS) Problem:** Cluster the logic cells to minimum number of clusters and find the optimum wake up times for logic clusters in the circuit so as to minimize the total turn-on time $T_{\text{TURNON}}$ while satisfying $I_{\text{TURNON}} < I_{\text{MAX}}$.

We propose an algorithm, called Wakeup Scheduler (WS), to solve the problem. The WS comprises of two steps:

1. **Logic Cell Cluster Generator:** We partition logic cells in the target circuit into a number of disjoint clusters $C_1, C_2, \ldots, C_M$ and assign exactly one sleep transistor with one sleep/wake up signal to all the cells in each cluster. The goal of clustering is to minimize the number of clusters, $M$, such that the total turn-on current flowing through the sleep transistors associated with each cluster, $\text{ITURNON}(C_i)$, does not exceed $I_{\text{MAX}}$.

2. **Inter-Cluster Sleep Signal Scheduler:** Consider a single sleep signal that drives the sleep transistor of a cluster. The goal of wake up signal scheduling is to provide the ordering and relative timing of the activation signals for the $M$ sleep signals in the circuit to minimize the overall wake up time while limiting the total current flowing to ground to $I_{\text{MAX}}$.

In this paper, we solve the WSS problem by solving each of the clustering and scheduling problems separately and sequentially. Based on the discussion in the previous section, since short circuit currents can be avoided by an appropriate turn-on strategy which in turn reduces the total $T_{\text{TURNON}}$, we do clustering and scheduling in a way that short circuit currents are eliminated. We construct a constraint graph to capture this requirement and use it during the clustering and scheduling steps. This graph captures the zero-short-circuit current requirement and is used during the logic cell clustering and sleep signal scheduling steps.

The constraint graph $G(V,E)$ is a Weighted Directed Acyclic Graph. Each vertex, $v_i$, in the graph corresponds to a logic cell in the circuit. There is a directed edge $e(v_i, v_j)$ from $v_i$ to $v_j$ if and only if $v_i$ is in the fanout of $v_j$ and the output of $v_i$ transitions from 1 to 0 during the circuit turn-on time under the specified sleep vector. In addition, there is a positive weight, $w(i, j) = T_{\text{SETTLE}}(v_j)$, associated with each edge $e(v_i, v_j)$, where $T_{\text{SETTLE}}(v_i)$ is the time required for the output of cell $v_i$ to settle to its final value when its sleep transistor is turned on. $T_{\text{SETTLE}}(v_j)$ values are calculated by circuit simulation as follows. If the primary sleep vector is known and each logic cell is turned on only after all its fanin cells are turned on, then the input values of the logic cell will be known at the time it is turned on. Therefore, we can simulate the cell under the specific sleep vector value in order to find $T_{\text{SETTLE}}(v_j)$ and the current profile of the cell (i.e., $I_{\text{TURNON}}(v_j)$) after its sleep transistor is turned on (i.e., SLEEP=0). To ensure that there will be no short circuit current, the following constraint on graph $G$ is enforced.

If nodes $v_i$ and $v_j$ are in the same cluster $C_k$, then there should be no directed path between $v_i$ and $v_j$ or vice versa.

Clearly, if there is an edge $e(v_i, v_j)$ where both $v_i$ and $v_j$ are in the same cluster, their corresponding cells will be turned on at the same time. However, output of node $v_i$ will be making a
downward transition. Hence, short circuit current will flow through cell $v_i$. (cf. discussion following Figure 4.) Note that this constraint implies that, there should not be any path between two nodes in a cluster going through nodes outside the cluster. This will be explained in more detail below. The constraint on the sleep signal scheduling step can be described more clearly if we define a new directed graph, $G_C$, called the Cluster Constraint Graph. The vertices of this graph correspond to the clusters $C_1, C_2, ..., C_M$. There is an edge from $C_i$ to $C_j$ in $G_C$ if and only if there is at least one edge from some node of $C_i$ to some node of $C_j$ in the original constraint graph $G$. There is a positive weight associated with each edge in $G_C$. The edge weight is calculated as follows: 

$$w(C_k, C_l) = \max\{w(v_i, v_j) | v_i \in C_k, v_j \in C_l\}.$$ 

Although $G$ is acyclic (assuming combinational logic circuits), there is no guarantee that a clustering solution will result in an acyclic $G_C$. An example is provided in Figure 8 where there is a cycle between clusters $C_k$ and $C_l$. Clearly, there is no way to schedule $C_k$ and $C_l$ to avoid short circuit current. If $C_l$ is turned on after $C_k$, there will be a cell $v_k \in C_l$ driving another cell $v_l \in C_k$ which is already on. Therefore, cell $v_l$ will consume short circuit current. A similar problem arises if $C_k$ is turned on first. Hence the Cluster Constraint Graph should be acyclic.

**Figure 8. An example of a cyclic constraint graph.**

The constraint imposed on the sleep signal scheduling by the presence of edge $e(C_k, C_l)$ in $G_C$ is: $T_{OM}(C_k) + w(C_k, C_l) \leq T_{OM}(C_l)$ where $T_{OM}(C_k)$ and $T_{OM}(C_l)$ are the turn-on times of clusters $C_k$ and $C_l$, respectively. Using the above information, we define the clustering problem as follows.

**Logic Cell Clustering (LCC) Problem:** Partition logic cells $v_1, v_2, ..., v_N$ into a minimum number, $M$, of clusters $C_1, C_2, ..., C_M$ such that there is no cycle in $G_C$ and $\max(I_{TURNON}(C_k)) \leq I_{MAX}$ for all $k$ where, $I_{TURNON}(C_k) = \sum_{v_i \in C_k} I_{TURNON}(v_i).$

where summation is point-wise and $MAX$ is taken over time. Note $I_{TURNON}(C_k)$ and $I_{TURNON}(V_i)$ represent the turn-on current waveforms, and not scalar current values.

### LCC Algorithm

1. **For** all cells $v_i$ in the circuit do  
2. **For** all clusters $C_k$ created so far do  
3. If adding $v_i$ to cluster $C_k$ creates a cycle in $G_C$ or violates the $I_{MAX}$ threshold for $C_k$  
4. Then continue with the next cluster (goto 2);  
5. Else {add $v_i$ to cluster $C_k$; update $G_C$; continue with the next cell (goto 1);}  
6. Create a new cluster and add $v_i$ to it; Update $G_C$;

Notice that we aim to minimize the number of clusters in order to reduce the number of sleep signals that are required in our proposed power gating structure. This will in turn simplify the power management circuitry. While using one sleep signal per cluster may seem costly, it is notable that in [7], a sleep signal tree (which is merely an inverter tree) similar to a clock tree has been proposed to drive large sleep transistors used in power gating structures. It is, therefore, possible to generate different timing for sleep signals going to different clusters by simply inserting delay elements (buffers) in the sleep signal tree.

**Sleep Signal Scheduling (SSS) Problem:** Determine $T_{OM}(C_k)$ values to minimize the total turn-on time subject to $\max(I_{TURNON}(C)) \leq I_{MAX}$ and $T_{OM}(C_k) + e(C_k, C_l) \leq T_{OM}(C_l)$ constraints.

For a given ordering of clusters, $T_{OM}(C_1) < \ldots < T_{OM}(C_k) < T_{OM}(C_{k+1}) < \ldots < T_{OM}(C_M)$, it may be possible to shift the current waveforms of two clusters $I_{TURNON}(C_k)$ and $I_{TURNON}(C_{k+1})$ to overlap one another without violating the constraint $I_{TURNON} \leq I_{MAX}$. The question is how close $T_{OM}(C_k)$ and $T_{OM}(C_{k+1})$ can be scheduled without violating the $I_{MAX}$ constraint. To address this problem, we augment $G_C$ with a new set of weighted directed edges $d(C_k, C_{k+1})$ as follows:

$$d(C_k, C_{k+1}) = \min\{\Delta T\} \quad \text{s.t.} \max[I_{TURNON}(C_k) + \text{shift}[I_{TURNON}(C_{k+1}), \Delta T]] \leq I_{MAX}$$  

where $\text{shift}[I_{TURNON}(C_{k+1}), \Delta T]$ is the waveform $I_{TURNON}(C_{k+1})$ shifted right on the time axis by an amount $\Delta T$ (Figure 9).

![Figure 9. Shifting clusters.](image)

Next, we combine edge weights $w(C_k, C_l)$ and $d(C_k, C_l)$ to construct new edge weights, which capture both constraints:

$$f(C_k, C_l) = \max\{w(C_k, C_l), d(C_k, C_l)\}.$$ 

For the given ordering of clusters, the minimum turn-on time can be described as

$$\sum_{k=1}^{M-1} \{f(C_k, C_{k+1}) + T_{SETTLE}(C_M)\},$$

which is the weight of a path in graph $G_C$ with edges $f(C_k, C_l)$ going through each vertex exactly once plus $T_{SETTLE}(C_0) = \max(T_{SETTLE} (v) | v \in C_0)$, where $T_{SETTLE}(V_i)$ is the time required for the output of cell $v_i$ to settle during the turn on time.

To consider the settling time of the last cluster we add a dummy vertex $C_0$ to the graph with no outgoing edges and the following incoming edges, $f(C_0, C_k) = T_{SETTLE}(C_k)$ for all $k$.

The WSS problem is restated as: “Find the minimum weighted directed Hamiltonian path on graph $G_C$ with edges $f(C_k, C_l)$.”

Recall that a Hamiltonian path is a path including all vertices of a graph exactly once [9]. Clearly a Hamiltonian path of the graph with a dummy node, $C_0$, should end at $C_M$ which has no outgoing edges. There are many heuristics for solving the minimum Hamiltonian path problem, which is an NP-complete problem [9]. However, because the number of clusters is usually small even for a large circuit, using an exhaustive search for solving the minimum Hamiltonian path is feasible. The scheduling step results in the optimal turn-on times, $T_{TURNON}(C_k)$ for a given ordering of clusters. If the number of clusters is small, it is possible to exhaustively try all possible orderings, and thereby, find the best ordering. Otherwise, an ordering of clusters can be arbitrarily or heuristically selected. One heuristic could be as follows: Select an arbitrary cluster $C_k$ as the first cluster to be scheduled to wake up. Next find the next cluster $C_{k+1}$ that minimizes $d(C_k, C_{k+1})$ (i.e., $d(C_k, C_{k+1}) \leq d(C_k, C_{k+2})$ for every $k$) and continue in the same way (i.e., at step $k$: $d(C_k, C_{k+1}) \leq d(C_k, C_{k+2})$).
In practice since the shape of current profile of clusters is very similar, the initial ordering used in our algorithm is not important. Note that in our method by changing the value of the maximum current bound, the wake up time can be reduced.

5. Input-driven Sleep Transistor Typing

In the previous section, we saw how the short circuit current can be avoided when turning on a circuit. Another approach to eliminating the short circuit current during the wakeup is to judiciously use an NMOS or a PMOS sleep transistor for each logic cell in the circuit (we call this technique Input-driven Sleep Transistor Typing, or ISTT for short.) The basic idea is that, for the given sleep vector, if the output of a logic cell in the circuit is logic 1, then an NMOS sleep transistor will be used to disconnect that cell from the ground; otherwise, a PMOS sleep transistor will be used to disconnect the output from \( V_{dd} \) as is shown in Figure 10.

With this simple ISTT algorithm, we ensure that every logic cell uses the type of the sleep transistor that minimizes the leakage current through the off-path of the logic cell through the well-known sleep transistor induced stack effect [8]. As a result, the output of every logic cell under the given sleep vector is driven to a \textit{hard zero} or one logic level. Therefore, no logic cell will have a floating output node (which would have resulted in intermediate signal values changing during the sleep mode thereby causing a potentially large short-circuit current during transition to the wakeup mode.) Furthermore, in this case, the only floating nodes in the circuit are some of the internal nodes of logic cells (e.g., the shared diffusion area between source of the NMOS driver transistor and drain of the NMOS sleep transistor in the first stage of the inverter chain of Figure 10). These internal floating nodes can change during the sleep mode, and therefore, there will be some current dissipation on wakeup time to recover their correct values. However, this current is significantly less than the current that will flow thru the circuit when only NMOS sleep transistors are used. The reason, is that in the latter case, not only some of the internal nodes of logic gates are floating, but also, on average, half of the output nodes of the logic cells (which typically drive larger internal nodes of logic gates are floating, but also, on average, half that will flow thru the circuit when only NMOS sleep transistors are used.) Situation is similar, the initial ordering used in our algorithm is not important.

6. Simulation Results

We used HSPICE to find the delay and current profile of each logic cell in a 0.09 \( \mu \)m standard cell library for a given sleep vector for all possible input combinations to the logic cell. Next we applied our algorithm to a number of circuits from the ISCAS test benchmark suite. Table 1 provides the amount of charge that is flowing to the ground during the wakeup time, and the amount of charge that is sourced from the power supply during the same time. We report results for the following techniques: a single (properly sized) NMOS device is used as the sleep transistor for the entire circuit (named Single-N), the two techniques of reference [5] (named Staircase-SS and Parallel-ST), the proposed WS (Wakeup Scheduler) with NMOS sleep transistors only (WS-N), the ISTT (Input-driven Sleep Transistor Typing), and finally the ISST+WS technique. For each technique two columns of data are reported; one includes the amount of charge flowing to ground and the other one is the amount of charge coming from supply voltage. The last two columns in the table provide the number of clusters for the WS and ISST+WS techniques. Note that for the latter, there are two types of clusters, one for the NMOS sleep transistors and one for the PMOS sleep transistors. Both data values are reported in the last column of Table 1 with a '+' separator. All data in the tables are generated with a 1.8V \( V_{dd} \). The clock cycle time for each circuit was set to the worst cast delay.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Charge Flowing to Gnd during Wakeup</th>
<th>Charge Coming from Vdd during Wakeup</th>
<th># Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single ST</td>
<td>Staircase SS</td>
<td>Parallel ST</td>
<td>WS</td>
</tr>
<tr>
<td>9sym</td>
<td>8.8</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>C432</td>
<td>5.5</td>
<td>0.7</td>
<td>5.4</td>
</tr>
<tr>
<td>C1355</td>
<td>10.7</td>
<td>1</td>
<td>10.3</td>
</tr>
<tr>
<td>C1908</td>
<td>24.9</td>
<td>7.1</td>
<td>21.2</td>
</tr>
<tr>
<td>C2670</td>
<td>36.0</td>
<td>10.4</td>
<td>30.0</td>
</tr>
<tr>
<td>C3540</td>
<td>55.0</td>
<td>22.0</td>
<td>48.0</td>
</tr>
<tr>
<td>C5315</td>
<td>49.5</td>
<td>9.7</td>
<td>48.5</td>
</tr>
<tr>
<td>C6288</td>
<td>83.3</td>
<td>23.8</td>
<td>68.4</td>
</tr>
<tr>
<td>C7552</td>
<td>116</td>
<td>41.6</td>
<td>94.8</td>
</tr>
</tbody>
</table>

Table 1. Charge sinked to Gnd or sourced from Vdd (pico Coulombs.)

We observe that the ground current dominates the supply current for techniques that use NMOS sleep transistor only. Situation is reversed when both NMOS and PMOS sleep transistors are in use. This is because in the former case a lot more nodes (that were incidentally charged up during the sleep time) will have to be discharged to Gnd to assume their correct values at the onset of the circuit wakeup. Compared to previous techniques (Single ST, Staircase SS, and Parallel ST), the WS technique reduces the amount of charge flowing to the ground during wakeup. Further reduction can be achieved by using a mixture of NMOS and PMOS transistors in the circuit to reduce the number of internal nodes that need to be discharged during the wake up time. This is seen by the significant reduction in the amount of charge that is flowing to the ground for ISTT and ISST+WS methods.

Table 2 shows the maximum current of the ground and supply lines for all of the above techniques. For Parallel-ST, we have used the worst case delay of the circuit as the period of a clock signal, which is then used to turn on the sleep transistors in multiple cycles. To make the comparisons meaningful, we set the \( I_{max} \) constraint for the WS algorithm to the best that is achieved by Parallel-ST and Staircase-SS. As one can see, our proposed techniques reduce the maximum current of ground more than any other technique. The \( I_{max} \) values for ISST are higher than that for WS or ISST+WS because we have not restricted the ground current for ISST.
In another experiment for Parallel-ST and Staircase-SS methods, we uniformly distributed the sleep signal arrival times within a single clock cycle (in data reported above we used multiple clock cycles per ref. [5]). Next, we measured the maximum ground current and report the product of this current and the single-cycle wakeup time. The results are reported in Table 4. Comparing the products of maximum ground current and wake up time of our method in Table 3 and those in Table 4, we conclude that our techniques maintain the advantage (between one and two orders of magnitude) over Staircase-SS and Parallel-ST techniques even when they are implemented in a single cycle by between one and two orders of magnitude. Note that the wakeup times reported in Table 4 were calculated as the summation of the time required to apply the wakeup signals and the time required for all nodes in the circuit to settle.

7. Conclusions

We introduced a new method for reducing the wake up time and maximum current flowing to ground for power gating structures. One of the proposed techniques is based on effectively clustering logic cells and scheduling wakeup signals for the clusters to achieve the mentioned objectives. The algorithms provided in this paper have low computational complexity and yet very effective. Experimental results for our methods showed between one and two orders of magnitude improvement in the amount of maximum current going to ground multiplied by the wake up time compared to the previous methods.

Acknowledgment

The authors would like to thank Thomas Sidle, the VP of Advanced CAD Technologies group at Fujitsu Labs. of America for his support of this project.

References