Distributed Sleep Transistor Network for Power Reduction

Changbo Long, *Student Member, IEEE,* and Lei He, *Member, IEEE*

***Abstract*—Sleep transistors are effective to reduce leakage power during standby modes. The cluster-based design was proposed to save sleep transistor area by clustering gates to minimize the simultaneous switching current per cluster and inserting a sleep transistor per cluster. In this paper, we propose a novel distributed sleep transistor network (DSTN), and show that DSTN is intrinsically better than the cluster-based design in terms of the sleep transistor area and circuit performance. We reveal properties of optimal DSTN designs, and then develop an efficient algorithm for gate level DSTN synthesis. The algorithm obtains DSTN designs with up to 70.7% sleep transistor area reduction compared to cluster-based designs. Furthermore, we present custom layout designs to verify the area reduction by DSTN.**

***Index Terms*—Clustering, low-control overhead, low-power design, low-voltage, performance, performance tradeoffs.**

I. Introduction

Lowering supply voltage is effective for power reduction because of the quadratic relationship between supply voltage and dynamic power consumption. To compensate the performance loss due to a lower supply voltage, transistor threshold voltage V_t has to be decreased as well, which causes exponential increase in the subthreshold leakage current [1]. To reduce leakage power, multithreshold CMOS (MTCMOS, see Fig. 1) has been proposed with low V_t blocks connected to ground through high V_t transistors named as sleep transistors [1]–[4]. The sleep transistor is turned on when the circuit is in the computational mode, and is turned off to cutoff the power supply in the standby mode for significant power reduction.

As shown in Fig. 1(b), a chip is composed of blocks, such as ALU, control units and functional units. The gates of blocks are connected to power supply through sleep transistors. These sleep transistors are controlled by control signals generated by a power management processor (PMP) [3] or distributed control logics. We denote the gates sharing the same control signal as a *module* in this paper, and devote our efforts to studying the design of the sleep transistors for a *module*.

Existing work can be traced back to [3], in which the size of the sleep transistor is manually decided and is about 3% of the cell area. The concept of automatically sizing sleep transistors is introduced in [5], where the current exclusive discharging patterns are taken into account to reduce the transistor size. As argued by Anis in [6], the centralized sleep transistor design in [3], [5] suffers from large interconnect resistances between distant blocks. Such resistance has to be compensated by extra large sleep transistor area. They proposed a cluster-based design structure, where each cluster, consisting of several gates, is accommodated by a sleep transistor separately. The size of the sleep transistor is determined by the current of the cluster. To reduce the size of sleep transistor, they have also proposed two clustering algorithms based on bin-packing and set-partitioning techniques, respectively. Both clustering techniques minimize the simultaneous switching current of clusters.

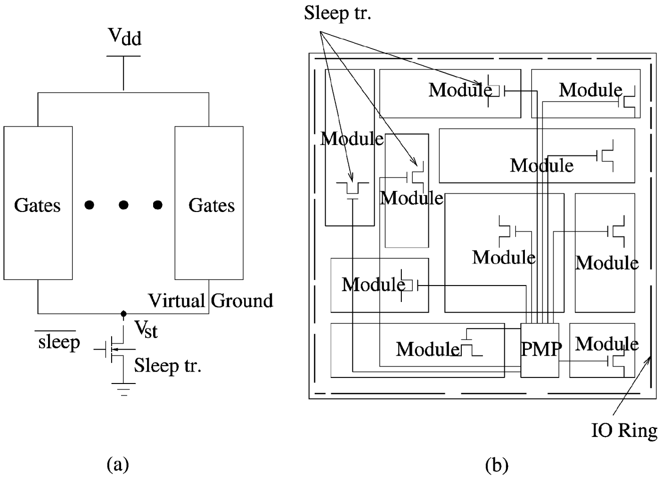


Fig. 1. Illustration of MTCMOS circuit structure and its application on the system level.

In this paper, we propose a novel distributed sleep transistor network (DSTN) with inherent advantages in area and performance compared to the cluster-based design. We discuss background knowledge in Section II, introduce the concept of DSTN in Section III, and propose a gate-level DSTN synthesis methodology in Section IV. We present experiments of gate-level synthesis and custom layout design in Section V and conclude in Section VI.

II. Background

A *module* is composed of *clusters*, which contains a number of gates. We denote the cluster-based sleep transistor design with a sleep transistor per cluster as CBSD, and use the term of module-base sleep transistor design to refer to the design style where large centralized sleep transistors are used for the entire module as in [3] and [5]. We also use *block* as a general term for both *module* and *cluster*.

Manuscript received September 10, 2003; revised March 12, 2004. This work was supported in part by the National Science Foundation (NSF) CAREER Award under CCR-0401682, in part by an SRC Grant under HJ-1008, and in part by a UC MICRO Grant sponsored by Analog Devices, Fujitsu Laboratories of America, Intel and LSI Logic, and a Faculty Partner Award by IBM.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90034 USA (e-mail: lhe@ee.ucla.edu).

Digital Object Identifier 10.1109/TVLSI.2004.832939

When sleep transistors are absent, the propagation delay for a CMOS gate can be approximated by

$$T_{\mathrm{pd}} \propto \frac{C_L V_{\mathrm{dd}}}{(V_{\mathrm{dd}} - V_{\mathrm{tL}})^{\alpha}} \tag{1}$$

where C_L is the load capacitance, V_{tL} is the threshold voltage in the low V_t module, and α is the velocity saturation index for modeling short channel effects [7]. When the sleep transistor is present and the source drain voltage drop is V_{st}, the gate propagation delay increases to

$$T_{\mathrm{pd-MT}} \propto \frac{C_L V_{\mathrm{dd}}}{(V_{\mathrm{dd}} - V_{\mathrm{st}} - v_{\mathrm{tL}})^{\alpha}}. \tag{2}$$

To measure the increase in propagation delay, performance loss (PL) is defined as in [6]

$$\mathrm{PL} = 1 - \frac{T_{\mathrm{pd}}}{T_{\mathrm{pd-MT}}}. \tag{3}$$

According to the analysis in [6], for $\mathrm{PL} = \delta$, we have

$$V_{\mathrm{st}} = \delta(V_{\mathrm{dd}} - V_{\mathrm{tL}}) \tag{4}$$

$$R_{\mathrm{st}} = \frac{\delta(V_{\mathrm{dd}} - V_{\mathrm{tL}})}{I_{\mathrm{st}}} \tag{5}$$

$$\left(\frac{W}{L}\right)_{\mathrm{st}} = \frac{I_{\mathrm{st}}}{\delta \mu_n C_{\mathrm{ox}} (V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})} \tag{6}$$

where I_{st} is the switching current in the low V_t module, V_{tH} is the threshold voltage of the sleep transistor and is higher than V_{tL} in the low V_t module (we assume $V_{\mathrm{tL}} = 350$ mV and $V_{\mathrm{tH}} = 500$ mV in this paper), and R_{st} is the channel resistance of the sleep transistor in the linear operation region.

Note that (W/L) is regarded as the area of the sleep transistor in this paper because transistors are implemented with minimum length in conventional designs. Moreover, the PL for blocks is assumed to be the same. In fact, certain blocks not in critical paths can tolerate a larger PL but not affect the overall PL of the circuit. How to leverage such property is our future work. Furthermore, to guarantee that the PL constraint holds for all possible input vectors, the maximum simultaneous switching current should be used as I_{st} in (6). Note that *Maximum simultaneous switching current (MSSC)* is the *worst-case current* generated by circuits [8]. It is also called *maximum envelope current* or *maximum instantaneous current* in the literature [9], [10].

Ignoring the resistance on the virtual ground due to the interconnects between sleep transistors and low-V_t gates (see Fig. 1), the total sleep transistor area of the module-based design and CBSD can be approximated as

$$\frac{\mathrm{MSSC_{mod}}}{\delta \mu_n C_{\mathrm{ox}} (V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})} \tag{7}$$

and

$$\frac{\sum_i \mathrm{MSSC_{clu_\mathit{i}}}}{\delta \mu_n C_{\mathrm{ox}} (V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})} \tag{8}$$

respectively, where $\mathrm{MSSC_{mod}}$ is the MSSC of a module, $\mathrm{MSSC_{clu}}$ is the MSSC of a cluster, and $\sum_i \mathrm{MSSC_{clu_\mathit{i}}}$ is the sum of $\mathrm{MSSC_{clu}}$ for all clusters. According to (7) and (8), the area of the module based design is smaller than that of CBSD because theoretically

$$\mathrm{MSSC_{mod}} \leq \sum_i \mathrm{MSSC_{clu_\mathit{i}}}. \tag{9}$$

$\mathrm{MSSC_{mod}}$ is actually much smaller than $\sum_i \mathrm{MSSC_{clu_\mathit{i}}}$ when the cluster size is much smaller than module size. For example, if we consider an extreme case in which every cluster contains only one gate, $\sum_i \mathrm{MSSC_{clu_\mathit{i}}}$ is the sum of the peak current for all gates, and $\mathrm{MSSC_{mod}}$ is the sum of peak current for those gates that simultaneously switch under a same input vector. Since only a small part of gates can switch simultaneously, $\mathrm{MSSC_{mod}}$ is much smaller than $\sum_i \mathrm{MSSC_{clu_\mathit{i}}}$. Therefore, the module-based design has a much smaller area compared to CBSD if the resistance on the virtual ground is ignored.

Considering the resistance on the virtual ground, the module-based design, however, leads to long virtual-ground wires and large resistance on the virtual-ground wires as pointed out in [6]. The increased resistance of virtual-ground wires has to be compensated by more area in the sleep transistor. In contrast, such overhead can be avoided by having a local sleep transistor per cluster, as in CBSD, and sleep transistor area can be further reduced by clustering the gates that do not switch simultaneously together to minimize the MSSC in the cluster. Note that the gates in a cluster should be placed adjacent to each other to minimize the length of virtual ground. On the other hand, timing-driven placement is aimed to placing gates with logic connections close to each other to minimize interconnect delay. Because gates with logic connections often have overlapped switching timing windows, there is a conflict between minimizing virtual ground length in CBSD and minimizing interconnect length in timing-driven placement.

In this paper, we will propose the DSTN design, and show that DSTN not only is compatible with timing-driven placement, but also has a reduced area for both sleep transistors and virtual-ground wires. Owing to the fact that the CBSD is better than the module-based design [6], we compare DSTN mainly with CBSD in the rest of the paper.

III. Distributed Sleep Transistor Network

Illustrated in Fig. 2(a) is the structure of CBSD, where gates in a cluster are connected to the sleep transistor by virtual-ground wires. The spot at which sleep transistor is connected to logic gates is called tapping point. By adding more wires to form a mesh containing all virtual-ground wires, we obtain the DSTN structure in Fig. 2(b).

Compared to CBSD, we show that DSTN reduces sleep transistor area from two reasons. First, according to Theorem 1 to be presented in Section IV, the total sleep transistor area of DSTN is equal to (7) when the interconnect resistance R_i between adjacent clusters is ignored. As R_i is much smaller than the resistance of the transistors, the increase in area is very small if considering R_i. Because the sleep transistor area of CBSD given by (8) is much larger than (7), the area of DSTN should be much smaller than that of CBSD.

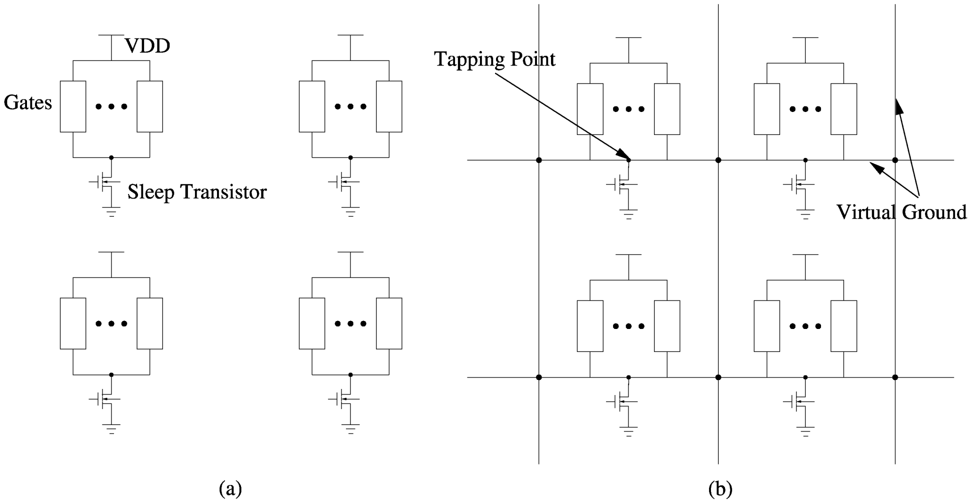


Fig. 2. (a) Cluster-based design. (b) Distributed sleep transistor network.

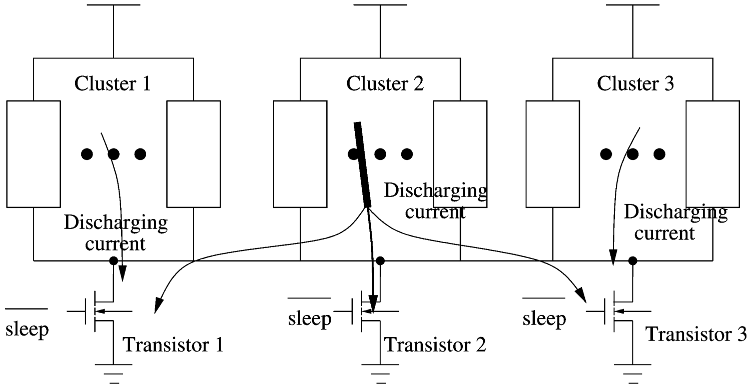


Fig. 3. Current discharging balance in DSTN.

Second, the area reduction can also be explained by the discharging current balance phenomenon. As shown in Fig. 3, the switching current in cluster 2 is larger than those in cluster 1 and cluster 3. When discharging current flows over sleep transistors, the voltage drop in sleep transistor 2 tends to be larger than the voltage drop in sleep transistor 1 and 3, which causes a part of the current from cluster 2 to flow to transistors 1 and 3[1]. Due to the discharging current balance phenomenon, the size of each transistor in DSTN never needs to be sized up to accommodate $\mathrm{MSSC}_{\mathrm{clu}}$, which in contrast exactly determines the size of transistors in CBSD.

Inserting sleep transistors introduces routing area overhead. Assuming the sleep transistors are connected to ideal ground, the overhead for DSTN, CBSD and module-based design is compared as follows. DSTN has a similar topology for virtual-ground wires with module-based design, but the wire size for DSTN is found to be much smaller due to the proximity of sleep transistors. Therefore, routing overhead for DSTN is much smaller than that of module-based design. On the other hand, DSTN needs more virtual-ground wires than CBSD. For comparison, we illustrate the routing topology for DSTN and CBSD in Fig. 4. The dotted lines are virtual-ground wires inside modules and are required by both DSTN and CBSD. Solid lines are

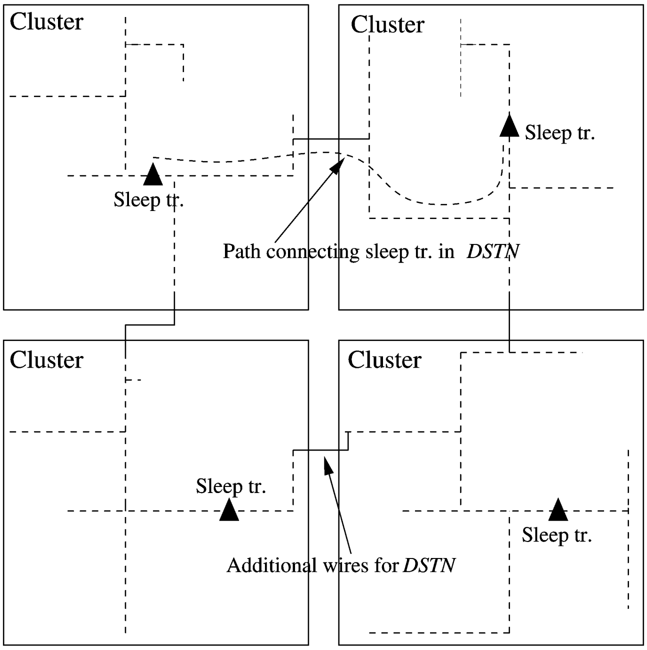


Fig. 4. Illusion of the virtual ground wires in CBSD and DSTN.

those virtual-ground wires only needed by DSTN. As shown in Fig. 4, these solid lines are short for compacted layout

[1] A similar discharging current balance has been discussed for P/G modeling [11].

designs. When the chip has a few "isolated" compacted layout regions such as IP-blocks in system-on-chip designs, we can simply apply individual DSTN inside each IP-block without introducing extra long virtual-ground wires. So, it is fair to say that the virtual ground length of DSTN is only slightly larger than that of CBSD.

Furthermore, introducing cluster methodologies in the sleep transistor design can affect placement. A good clustering solution minimizing the $\mathrm{MSSC_{clu}}$ is crucial to reducing sleep transistor area in CBSD. Such clustering helps DSTN as well. However, our experiments to be presented show that DSTN without $\mathrm{MSSC_{clu}}$ minimization achieves significant sleep transistor area reduction compared to CBSD with $\mathrm{MSSC_{clu}}$ minimization. Due to the adverse effect of MSSC minimization on timing-driven placement, we suggest *not* applying $\mathrm{MSSC_{clu}}$ minimization to DSTN.

IV. Gate Level DSTN Synthesis

Under the DSTN design methodology, it is critical to size each sleep transistor properly to reduce the total transistor area while satisfying the given PL constraint. In this section, we first discuss the modeling of DSTN and then formulate and solve the DSTN sizing problem.

A. Modeling of DSTN

We model DSTN as a resistance network shown in Fig. 5 with resistance R_{st} for sleep transistors and R_i for virtual-ground wires. Note that R_i is necessary to accurately model the phenomenon of discharge current balance. Estimating R_i accurately, however, requires detailed layout information. Because the layout information is unavailable in the gate-level, we estimate R_i approximately and assume that R_i is uniform for the virtual ground wires between adjacent clusters. Specifically, we assume that the wire resistance is 0.05 Ω/μm, and the length of virtual-ground wires is 200 μm, i.e., $R_i = 10\ \Omega$. Given the assumption that each cluster has about six gates (decided by the typical sleep transistor size in Section 4.2), 200 μm is a conservative wire length for the virtual-ground between adjacent clusters.

How to model the current generated by clusters is critical to the DSTN sizing problem. Conventionally, the current can be modeled as either time-invariant [14] or time-variant variable [15]. When it is modeled as time-invariant variable, the maximum simultaneous switching current, i.e., $\mathrm{MSSC_{clu}}$ should be used for sleep transistor sizing to guarantee that the voltage drop constraint is never violated in the worst scenario. Time-invariant model may cause over-sizing of the sleep transistors because it ignores the interdependence between clusters and simply assume that $\mathrm{MSSC_{clu}}$ for clusters happens simultaneously. To avoid over-sizing, the current model with consideration of time variant and interdependence are employed in this paper. The switching current of cluster i is denoted as $I_i(v,t)$ (see Fig. 5), showing that the current is a variable w.r.t. input vector v and time t. In practice, the switching current of a gate is modeled by piece-wise linear waveforms obtained through circuit simulation (See Section V).

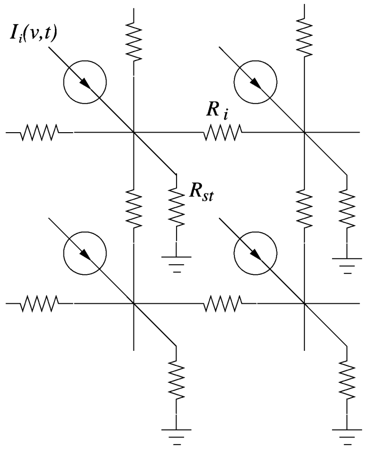


Fig. 5. Resistance network modeling of DSTN.

B. DSTN Sizing

1) Problem Formulation: DSTN sizing problem is formulated as follows.

Formulation 1: DSTN sizing problem (DSTN/SP): Given DSTN topology, DSTN/SP finds a size for every sleep transistor such that the total transistor area of DSTN is minimized and the PL constraint is satisfied for every cluster.

DSTN/SP is more difficult than the sizing problem in CBSD. According to (6), the size of every sleep transistor in CBSD is solely determined by $\mathrm{MSSC_{clu}}$. Owing to discharging current balance in DSTN, the size of a sleep transistor in DSTN depends on the total amount current going through the directly connected cluster, the adjacent clusters, and even nonadjacent clusters.

Nevertheless, efficient sizing algorithms can be found for a general resistance network [14]–[18], by which DSTN is modeled. Most of these algorithms are originally developed to size P/G network. However, the resistance network of DSTN distinguishes itself by special properties. Based on these properties, well-designed heuristics may lead to good solutions as well but in a more efficient fashion. In the following, we reveal these properties first and then propose our heuristic sizing algorithm. For simplicity of presentation, we call the total area of sleep transistors in DSTN as the total area of DSTN in the rest of the paper.

2) Properties: A critical observation to the resistance network of DSTN is that R_i is much smaller than R_{st}. For example, the channel resistance of the transistor in the linear-operation region is

$$R_{\mathrm{st}} = \frac{1}{\mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tH}})}\left(\frac{L}{W}\right). \tag{10}$$

We assume $V_{\mathrm{tH}} = 500$ mV, a typical sleep transistor in DSTN has $(W/L) = 6$, and $V_{\mathrm{dd}} = 1.3$ V in 100 nm technology. The typical resistance value for R_{st} is around 218 Ω. On the other hand, a 200 μm-long virtual-ground wire has R_i of about 10 Ω in 100 nm technology. Therefore, it is reasonable to assume that R_{st} is much larger then R_i.

Let $\mathbf{A} = \sum_i (W/L)_{\mathrm{st}_i}$ be the total area of sleep transistors, and $\mathbf{A}^*$ be the minimum area satisfying a given voltage drop constraint. We reveal the properties for DSTN sizing as follows.

Theorem 1: Assuming $R_i = 0$ and $\mathrm{PL} = \delta$

$$\mathbf{A}^* = \frac{\mathrm{MSSC}_{\mathrm{mod}}}{\delta \mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})}. \tag{11}$$

Proof: When $R_i = 0$, all sleep transistors in DSTN can be viewed as one single transistor with channel resistance and area of

$$R = \frac{1}{\sum_i 1/R_{\mathrm{st}_i}} \tag{12}$$

$$\mathbf{A} = \sum_i \left(\frac{W}{L}\right)_{\mathrm{st}_i}. \tag{13}$$

To satisfy the voltage drop constraint, we have

$$\mathbf{A} \geq \frac{\mathrm{MSSC}_{\mathrm{mod}}}{\delta \mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})}. \tag{14}$$

Therefore

$$\mathbf{A}^* = \frac{\mathrm{MSSC}_{\mathrm{mod}}}{\delta \mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})}. \tag{15}$$

□

Theorem 2: Given PL as a constant, $\mathbf{A}^*$ never decreases when R_i increases.

Proof: For the purpose of contradiction, we assume

$$R_i^{(1)} > R_i^{(2)} \tag{16}$$

$$\mathbf{A}^{*(1)} < \mathbf{A}^{*(2)} \tag{17}$$

where $\mathbf{A}^{*(1)}$ is the minimum area of sleep transistors when all interconnect resistors are $R_i^{(1)}$ (case 1), and $\mathbf{A}^{*(2)}$ is that when all these resistors are $R_i^{(2)}$ (case 2). For case 1, assuming the interconnect resistors is decreased from $R_i^{(1)}$ to $R_i^{(2)}$, the voltages of the tapping points never increase and the performance loss constraint is therefore still satisfied. In this way, we end up with case 3 in which the interconnect resistance is $R_i^{(2)}$ and the area of sleep transistors is $\mathbf{A}^{*(1)}$. However, the existence of case 3 violates the assumption that $\mathbf{A}^{*(2)}$ is the minimum area when the interconnect resistors are $R_i^{(2)}$ because

$$\mathbf{A}^{*(1)} < \mathbf{A}^{*(2)}. \tag{18}$$

Therefore, when

$$R_i^{(1)} > R_i^{(2)}, \tag{19}$$

we must have

$$\mathbf{A}^{*(1)} \geq \mathbf{A}^{*(2)}. \tag{20}$$

□

Lemma 1: Let I_i be the switching current of cluster c_i at time t under vector v and $\overline{\mathrm{PL}}$ be the maximum PL among all tapping points, we have

$$\overline{\mathrm{PL}} \geq \frac{\sum_i I_i}{\mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}}) \cdot \mathbf{A}}. \tag{21}$$

Proof: For the purpose of contradiction, we assume

$$\mathrm{PL}_i < \frac{\sum_i I_i}{\mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}}) \cdot \mathbf{A}} \tag{22}$$

holds for all i. According to (6)

$$I_i' = PL_i \cdot \mathbf{A}_i \cdot \mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}}) \tag{23}$$

where $\mathbf{A}_i$ and I_i' is the area and the current flowing sleep transistor st_i, respectively. Hence

$$\sum_i I_i' = \sum_i \mathrm{PL}_i \cdot (\mathbf{A}_i \cdot \mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}}) \tag{24}$$

$$< \frac{\sum_i I_i}{\mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}}) \cdot \mathbf{A}} \tag{25}$$

$$\cdot \sum_i \{\mathbf{A}_i \cdot \mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}})\} \tag{26}$$

$$< \sum_i I_i. \tag{27}$$

However, according to the Kirchoff's Current Law (KCL)

$$\sum_i I_i' = \sum_i I_i \tag{28}$$

which is contradict to (27). Therefore, inequality (21) must hold. □

Theorem 3: For a particular time t under certain input vector v and given the total area of sleep transistors $\mathbf{A}$, the following

$$\mathbf{A}_i = \mathbf{A} \cdot \frac{I_i}{\sum_i I_i} \tag{29}$$

lead to minimized $\overline{\mathrm{PL}}$. Note that I_i is the current of cluster c_i in t under v and $\mathbf{A}_i$ is the area of the corresponding sleep transistor st_i.

Proof: When

$$\mathbf{A}_i = \mathbf{A} \cdot \frac{I_i}{\sum_i I_i} \tag{30}$$

PL_i are uniform and equal to

$$\frac{\sum_i I_i}{\mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}} - V_{\mathrm{tL}})(V_{\mathrm{dd}} - V_{\mathrm{tH}}) \cdot \mathbf{A}}. \tag{31}$$

According to Lemma 1, $\overline{\mathrm{PL}}$ is minimized. □

The total area of DSTN can be roughly determined by combining Theorems 1 and 2. If $R_i = 0$, the total area of the DSTN is given by (15). However, according to Theorem 2, the total transistor area in DSTN must be larger than the value in (15). Nevertheless, the effective resistance increase at the tapping point is limited because R_i is much smaller than R_{st}. The increase of transistor area in DSTN is therefore limited.

Theorem 3 shows that to minimize the maximum PL, the area should be assigned to each individual sleep transistor proportionally to current of the cluster to which it is connected. However, the switching current of the clusters vary with time and input vector, and there exists no exact constant ratio between them. Nevertheless, some constant value, such as $\mathrm{MSSC}_{\mathrm{clu}}$ and average current, serves this purpose well because they represent the common cases. In this paper, we employ $\mathrm{MSSC}_{\mathrm{clu}}$ as the criteria to assign area to each individual sleep transistors in DSTN.

3) Sizing Algorithm: The sizing algorithm is described as follows. We first calculate $\mathrm{MSSC}_{\mathrm{mod}}$ and then calculate the total area of DSTN according to the following formula:

$$\mathbf{A} = (1+\beta)\cdot\frac{\mathrm{MSSC}_{\mathrm{mod}}}{\delta\mu_n C_{\mathrm{ox}}(V_{\mathrm{dd}}-V_{\mathrm{tL}})(V_{\mathrm{dd}}-V_{\mathrm{tH}})}. \quad (32)$$

β is an empirical parameter to consider the effect of R_i. Because the effect of R_i becomes more significant as the size of the circuit increases, we empirically employ the following formula to calculate β

$$\beta = 0.002\cdot N_{\mathrm{clu}} \quad (33)$$

where N_{clu} is the number of cluster in the circuit. In addition, a bigger β may be used if the value of R_i increases. Finally, according to Theorem 3, the total DSTN area is allocated to each sleep transistor st_i proportionally to the correspondent $\mathrm{MSSC}_{\mathrm{clu}}$, i.e.,

$$\mathbf{A}_i = \mathbf{A}\cdot\frac{\mathrm{MSSC}_{\mathrm{clu}_i}}{\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}}. \quad (34)$$

C. MSSC Calculation

1) $\mathrm{MSSC}_{\mathrm{mod}}$*:* The estimation for maximum current through the supply lines has been well studied due to its criticality to the reliability of the power/ground network [8], [10], [11], [19]–[21]. Most of these methods can be adopted to estimate the maximum current through the sleep transistors because they are primarily based on estimating $\mathrm{MSSC}_{\mathrm{mod}}$. We employ Genetic algorithm (GA) based algorithm to calculate $\mathrm{MSSC}_{\mathrm{mod}}$ in this paper because of its high accuracy and scalable running time.

The basic idea of GA-base $\mathrm{MSSC}_{\mathrm{mod}}$ estimation is to find the input vector which generates maximum current in the circuit by fitting the optimization process into a GA framework. Primary input signals are coded as 00, 11, 01, and 10, respectively, which represent the four types of input signal of 0, 1, rising and falling. The codes for all primary inputs constitute a string representing an input vector. Each string is associated with a fitness value, i.e., the maximum current under the input vector. A group of strings is a generation, which transforms via the operations of selection, crossover and mutation. As the transform proceeds, the fitness value of strings becomes increasingly closer to the optimum value, and the process terminates when the fitness value gets close enough to the optimum value.

The scheme for evolution options and termination criteria depends on the application context, and is critical to the quality of the result. We use the schemes from [10]. The current model used in our work will be discussed in the Appendix.

2) $\mathrm{MSSC}_{\mathrm{clu}}$*:* GA based algorithm can be applied to calculate $\mathrm{MSSC}_{\mathrm{clu}}$ as well, i.e., for each cluster, a GA process is performed to find an input vector that generates the maximum current for a given cluster. However, it is time-consuming when the number of clusters is big for a large circuit. We will present an efficient method to estimate an upper bound of $\mathrm{MSSC}_{\mathrm{clu}}$ in the Appendix. Nevertheless the proposed DSTN sizing algorithm works with any MSSC estimation method.

D. Comparison Base

We employ CBSD as the comparison base in the experiment. According to (8), the total area of sleep transistors for CBSD is proportional to $\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}$. However, $\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}$ strongly depends on how to group gates together to form clusters. Clustering algorithms with and without placement constraints for reducing $\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}$ have been proposed in [6]. In general, the sleep transistor area without considering placement constraint is smaller than the area considering placement constraint. For strict comparison, we compare DSTN with CBSD without considering placement constraints. The problem is formulated as follows.

Formulation 2: CBSD without placement constraints (CBSD/P): Given a module and cluster size, group gates into clusters such that $\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}$ is minimized and the total area of sleep transistors is minimized.

To reach the maximum potential of sleep transistor area reduction, we apply simulated annealing (SA) to tackle the CBSD/P problem. In SA, each cluster is associated with a cost of $\mathrm{MSSC}_{\mathrm{clu}}$ and total cost is $\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}$. SA starts with a random clustering. In each move, two gates are randomly picked from two arbitrary clusters and exchanged. We start SA with temperature of 100 and terminate at 0.1. Temperature decreases rate is 0.9. The number of moves at a particular temperature is 200× of the number of clusters.

V. Experiment Results and Discussions

A. Gate Level Synthesis

All proposed algorithms have been implemented inside SIS[22] environment. We use ISCAS benchmark circuits and report experiment results in Table I. A gate-level simulator has also been implemented to calculate voltages and current waveforms. Parameters needed to simulate a circuit, such as gate delay, loading capacitance, and switching current, are all extracted from SPICE simulations and built into tables. Simulation results from our simulator are within 20% difference from SPICE simulations, but it is much faster than the SPICE simulation. This simulator was used to verify the gate level synthesis in this section.

We first compare the area (i.e., transistor width) used by DSTN and CBSD/P. We measure area by the total channel width of sleep transistors. One can see that DSTN uses significantly smaller area than CBSD/P does. On average, the area reduction is 49.8%. Because we do not consider the delay constraint during placement for CBSD/P, we obtain a lower bound of $\mathrm{MSSC}_{\mathrm{clu}}$ in a timing-driven placement and a lower bound of the sleep transistor area in CBSD. Therefore, the area reduction by DSTN would be larger compared to CBSD if considering practical placement constraints.

We then compare performance loss. We have applied extensive random simulations to verify the quality for both sizing schemes. Because we consider only combinational circuits in the experiment, we apply random input with length of 10 000 clock cycles for each circuit to obtain the maximum PL (in

TABLE I
AREA AND MPL FOR DSTN AND CBSD/P (*CIRCUIT GATE NUMBER AFTER MAPPING IN SIS)

Circuit	#Gate*	#PI	#PO	Area (W/L)			$MPL(\%)$	
				$CBSD/P$	$DSTN$	Reduction (%)	$CBSD/P$ (lower bound)	$DSTN$
C432	323	36	7	439	205	53.3	7.04	3.80
C499	640	41	32	929	533	42.6	7.69	3.65
C880	528	60	26	801	581	27.5	6.25	3.42
C1355	625	41	32	878	532	39.4	7.53	3.05
C1908	830	33	25	1286	416	67.7	7.01	4.02
C2670	1459	233	140	1951	789	59.6	6.87	2.56
C3540	1613	50	22	2715	796	70.7	8.95	3.17
C5315	2813	178	123	4659	2302	50.6	7.62	2.64
C6288	2464	32	32	6219	3640	41.5	5.18	4.52
C7552	3685	207	108	6156	3377	45.1	5.16	3.96
Avg.	–	–	–	–	–	49.8	6.93	3.95

short, MPL). The performance loss for DSTN under simulations is computed as follows. For each clock, we first divide the clock cycle into ten segments. Because the P/G wire resistance is much smaller than the channel resistance of transistors, we assume that the performance loss is maximized at the same time as the maximum current happens within each segment. The purpose of dividing the switching window into ten segments is to obtain a solution close enough to the exact one.[2] PL for one clock cycle is the maximum one among all ten segments. Note that we compute performance loss by solving DSTN resistance network with a sparse linear equation solver integrated with SIS. The transistor channel resistance is computed by (10), and resistance for virtual-ground wires R_i is 10 Ω. MPL is the maximum performance loss among all 10 000 cycles. The same random simulations have been applied to calculate MPL in CBSD/P, where PL is calculated via (6). Under a particular simulation, maximum current of each cluster is found and this current is used to compute the performance loss of the cluster. The performance of the circuit is the maximum one among all clusters and MPL is the maximum performance loss among all 10 000 simulations.

As shown in Table I, MPL of DSTN satisfies the design constraint of 5% for all circuits and it is much smaller than that of CBSD/P. The performance loss violation of 5% in CBSD/P mainly comes from the underestimation of maximum current for clusters.

B. Custom Layout Design

The exact evaluation of most parameters, such as PL and transistor area, can only be obtained after a layout design. Therefore, we implement and compare three layout designs, sleep transistor free (ST-free) design, CBSD and DSTN, for a 4-bit carry-lookahead (CLA) adder.

The three layout designs are implemented as follows. First, a ST-free layout, consisting of four sum modules and one CLA module but without sleep transistors is implemented. Then, a CBSD layout is implemented by partitioning each module into 2–3 clusters and accommodating each cluster by one sleep transistor. Sleep transistor sizes are determined by SPICE simulations to keep PL below 5%. Finally, we implement a DSTN design by accommodating the entire CLA adder via six distributed sleep transistors. All these sleep transistors are connected together and have the same size[3]. As in CBSD, sizes of the sleep transistors in DSTN are determined by SPICE simulations to make PL below 5%.

TABLE II
LAYOUT DESIGN COMPARISON

Properties	ST-free	$CBSD$	$DSTN$
Leakage(nA)	59.80	5.72	1.23
Critical path delay(nS)	1.66	1.79	1.68
ST area(μm^2)	0	1449.6	212.2
Chip area(μm^2)	11960.0	13892.0	12880.0

As shown in Table II, compared to the ST-free design, both CBSD and DSTN achieve significant leakage current reduction but DSTN is approximately five times better than CBSD, which is mainly because the area in DSTN is several times smaller. Also, both CBSD and DSTN increase the critical path delay but DSTN has a much smaller delay than CBSD. These comparisons are consistent with previous theoretical analysis and experiment results.

VI. CONCLUSION AND FUTURE WORK

Sleep transistors are effective to reduce leakage power during standby modes. We have proposed a novel distributed sleep transistor network (DSTN), and have convincingly illustrated that DSTN has reduced area, less supply voltage drop, and no conflict with timing-driven placement when compared to existing module-based and cluster-based sleep transistor structures. We have revealed several properties of the optimal solution to the DSTN sizing problem, and have proposed an effective and efficient DSTN sizing algorithm based on these properties. Based on the experimental comparison with a rigorous cluster-based design, DSTN assuming conservative virtual-ground wires achieves on average 49.8% sleep transistor area reduction and leads to less performance loss. DSTN with these advantages can be used to implement power gating for

[2] In essence, the time step for our simulation is 10% of the clock cycle. Theoretically, the smaller the time step, the higher the simulation accuracy. However, a time step smaller than the one we used does not lead to a higher accuracy in our experiment.

[3] Same size is used because transistor area optimization techniques for individual sleep transistors make little difference in this small circuit.

reducing leakage power. An example of the system-level power gating scheduling has been discussed in [23].

The performance loss of logic gates is directly related to the ground bounce in DSTN, which is equivalent to the reduction of noise margin of logic gates. The $V_{\rm dd}$ level should be high enough to compensate such ground bounce and noise margin reduction. Ground bounce with presence of sleep transistors has been studies in [12], [13] and the principles can be extended to DSTN.

Sleep transistors can be viewed as a natural part of the power/ground network. We assume that the power/ground network (both global and virtual) is given *a priori* in this study. Our ongoing study has investigated the codesign of DSTN and power–ground network [24].

Appendix

In this Appendix, an efficient simulation-based algorithm to compute $\mathrm{MSSC_{clu}}$ is presented along with the experiment results to demonstrate the quality of the algorithm. Note that our sizing algorithm presented in Section IV can apply any current calculation algorithm.

A. Algorithm for Computing $\mathrm{MSSC_{clu}}$

1) Motivation: Applying GA to each cluster for obtaining $\mathrm{MSSC_{clu}}$ has a high cost because GA requires a large number of simulations and the number of clusters increases with the size of a circuit. In this paper, we present a simulation-based method, which first simulates the circuit under a small number of random input vectors and then estimates $\mathrm{MSSC_{clu}}$ for all clusters based on these simulations only. We denote a current waveform of these simulations as w and the set of these waveforms as $\mathbf{W}$. Based on the maximum current among all $w \in \mathbf{W}$, we propose to estimate $\mathrm{MSSC_{clu}}$ accurately by exploiting the correlation of switching between gates.

The correlation of switching between gates are commonly observed in a circuit. For example, the output of an *inverter* must switch if the input switches[4]. To illustrate the idea of exploiting the correlation of switching between gates, we present the following example. Let cluster c contains eight gates of $\{g_1, g_2, \ldots, g_8\}$. Assume at time t under a particular input vector g_3, g_4, g_7, and g_8 switch. These four switching gates build up six pairs: $(g_3, g_4), (g_3, g_7), \ldots, (g_7, g_8)$. A nonswitching gate, say g_1, is said to be correlated to switch with the pair of (g_3, g_4) if we can find some waveform $w \in \mathbf{W}$ in which g_1, g_3, and g_4 all switch. If a nonswitching gate is correlated to switches of all these six pairs it is artificially set to switch.

In essence, this heuristic is based on the following approximation:

$$g \sim G \approx \prod_{i,j} g \sim (g_i, g_j), \quad \forall g_i, g_j \in G \tag{35}$$

where symbol $\sim$ stands for simultaneous switching. The value of $g \sim G$ is 1 if g happens to switch simultaneously with all gates in G. Otherwise it is 0. Similarly $g \sim (g_i, g_j)$ is 1 if and only if g switch simultaneously with g_i and g_j. In above example, $G = \{g_3, g_4, g_7, g_8\}$. To obtain the value of $g \sim G$, we find out $g_1 \sim (g_3, g_4), \ldots, g_1 \sim (g_7, g_8)$ first and then multiply them together. Note that $g_1 \sim G = 1$ means that g_1 switch simultaneously with all gates in G. Therefore, g_1 is artificially set to 1 if $g_1 \sim (g_3, g_4), \ldots, g_1 \sim (g_7, g_8)$ are all 1. It can be imaged that we can employ the approximation of

$$g \sim G \approx \prod_{i,j,k} g \sim (g_i, g_j, g_k), \quad \forall g_i, g_j, g_k \in G \tag{36}$$

to improve the accuracy. Because it increases the computation cost significantly we employ (35) in this paper.

$$\mathbf{S} = \begin{array}{cccc|c} g_1 & g_2 & \dots & g_n & \\ \hline 1 & 0 & \dots & 1 & v_1 \\ 0 & 1 & \dots & 1 & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 0 & v_m \end{array}$$

Fig. 6. Definition of switching matrix.

```
Modify (S)
For x=1:n
   For j=1:m
      If S(x, i) = 1
         continue
      End if

      For all j and k satisfying S(x, j) = S(x, k) = 1
         If no y satisfying S(y, i) = S(y, j) = S(y, k) = 1
            continue
         End if
      End for

      S(x, i) = 1
   End for
End for
```

Fig. 7. Algorithm for modifying $\mathbf{S}$.

2) Details of the Algorithm: The heuristic proposed in Section A.1 is employed to estimate maximum current at a fixed time t. In order to estimate the maximum current for all time, we choose a few time and take the maximum value among them. Specifically, for each simulation $w_i \in \mathbf{W}$, we first find the time t at which current reaches the peak value. Then, we sort all these time points by the associated peak current in a decent order and estimate maximum current by the proposed heuristic in Section A-1 for the first ten time points. The maximum current $\mathrm{MSSC_{clu}}$ is the largest one among them. In the rest part of this section, we discuss the details of the algorithm to estimate maximum current at a particular time t, i.e., $\mathrm{MSSC}^t_{\mathrm{clu}}$.

We employ a matrix, denoted as *switching matrix* $(\mathbf{S})$, to represent the profile of switching activity for $\mathbf{W}$ at a particular time t. As shown in Fig. 6, the columns of $\mathbf{S}$ stands for gates in the cluster and the row of $\mathbf{S}$ stands for the input vectors.

To compute $\mathrm{MSSC}^t_{\mathrm{clu}}$, we take advantage of correlation of switching between gates to artificially set some nonswitching gates to switch. It is accomplished by modifying $\mathbf{S}$ according to the algorithm in Fig. 7. As shown in the algorithm, we process all the nonswitching entries in $\mathbf{S}$ one by one. Assume gate g_i is nonswitching but both gates g_j and g_k are switching under simulation w_x, i.e., $\mathbf{S}(\mathrm{x,i}) = 0, \mathbf{S}(\mathrm{x,j}) = 1$, and $\mathbf{S}(\mathrm{x,k}) = 1$. We attempt to find another vector w_y where all of them are switching. I.e., $\mathbf{S}(\mathrm{y,i}) = \mathbf{S}(\mathrm{y,j}) = \mathbf{S}(\mathrm{y,k}) = 1$. If there exists

[4] We say a gate switches if its output switches.

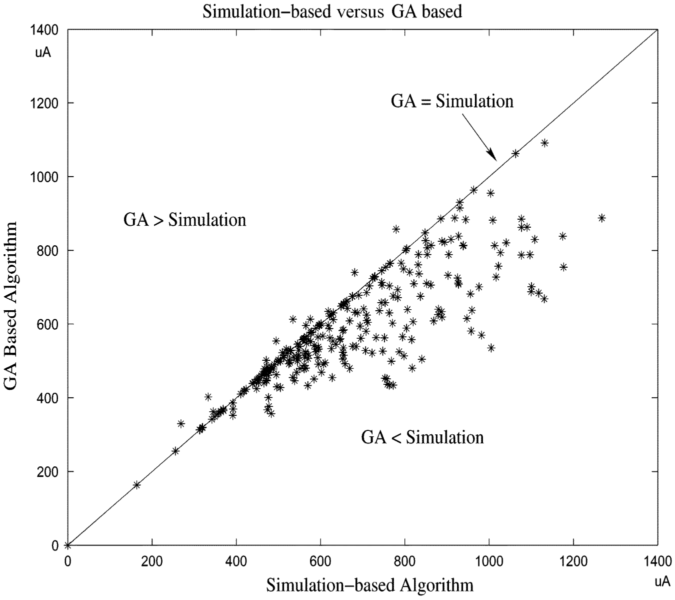


Fig. 8. Maximum current estimation by simulation-based algorithm versus by GA based algorithm.

no such w_y in $\mathbf{W}$, g_i is not correlated to switch with g_j and g_k. Therefore, g_i stay unchanged. If such w_y exists, we repeat this process for all pairs of switching gates in w_x. If such vector w_y exists for each pair, g_i is correlated to switch with all the switching gates under w_x and we artificially set it to switch. i.e., $\mathbf{S}(\mathrm{i},\mathrm{x})$ is modified to 1.

Modification of $\mathbf{S}$ is iteratively executed in our algorithm. It terminates when no changes can be made to $\mathbf{S}$. We denote the switching matrix after modification as $\mathbf{S}'$ and compute $\mathrm{MSSC}^t_{\mathrm{clu}}$ based on $\mathbf{S}'$

$$\mathrm{MSSC}^t_{\mathrm{clu}} = \max\left(\begin{bmatrix} \mathbf{S}'(1,1) & \cdots & \mathbf{S}'(1,n) \\ \vdots & \vdots & \vdots \\ \mathbf{S}'(m,1) & \cdots & \mathbf{S}'(m,n) \end{bmatrix} \begin{bmatrix} \bar{I}_{g_1}(t) \\ \vdots \\ \bar{I}_{g_n}(t) \end{bmatrix}\right) \tag{37}$$

where max returns the maximum value in a vector and

$$\bar{I}_{g_j}(t) = \max\left(\begin{bmatrix} I_{g_j}(v_1,t) \\ \vdots \\ I_{g_j}(v_n,t) \end{bmatrix}\right). \tag{38}$$

Note that $I_{g_j}(v_i,t)$ is the switching current of gate g_j at time t under input vector w_i.

Suppose $\mathbf{S}$ covers all possible input vectors, we have

$$\mathrm{MSSC}^{t*}_{\mathrm{clu}} = \max_{i=1}^{m}\left\{\sum_{j=1}^{n} \mathbf{S}(i,j)\cdot I_{g_j}(v_i,t)\right\} \tag{39}$$

where $\mathrm{MSSC}^{t*}_{\mathrm{clu}}$ is the ideal value of maximum switching current at t. According to (37) and (38), we have

$$\mathrm{MSSC}^t_{\mathrm{clu}} \geq \mathrm{MSSC}^{t*}_{\mathrm{clu}}. \tag{40}$$

Therefore, $\mathrm{MSSC}^t_{\mathrm{clu}}$ obtained by this simulation-based algorithm is actually an upper bound of the ideal value when $\mathbf{S}$ covers all input vectors. In Appendix B, we will use experiments to show that relatively small number of input vectors can achieve an upper bound of the ideal value in most cases.

TABLE III
COMPARISON OF RUNNING TIME

Circuit	# Gate	# Cluster	GA based	Sim.-based	Speed-up
		(s/cluster)	(s/cluster)		
C432	323	36	39.22	1.73	22.7x
C499	640	71	172.75	3.09	55.9x
C880	528	59	63.61	2.75	23.1x
C1355	625	70	147.87	2.73	54.2x
C1908	830	93	191.14	4.02	47.5x
C2670	1459	162	185.58	6.28	29.6x
C3540	1613	180	575.18	8.10	71.0x
C5315	2813	313	368.51	6.41	57.5x
C6288	2464	274	2357.64	6.17	382.1x
C7552	3685	410	683.77	8.77	78.0x

B. Experiment Result

As mentioned before, GA-based algorithm can also be applied to estimate the maximum current of clusters, which is similar to the GA-based algorithm for estimating the maximum current for the whole circuit. Because GA-based algorithm has shown high accuracy in [10], we employ it as a comparison base in the experiment. In fact, under the special context of sleep transistor sizing, of main concern of estimating $\mathrm{MSSC}^t_{\mathrm{clu}}$ is efficiency instead of accuracy. The reason is that in our sizing algorithm, $\mathrm{MSSC}_{\mathrm{clu}}$ is the coefficient to assign area to sleep transistors [see (34)], and an acceptable accuracy of the estimation is good enough due to the ability to balance switching current between sleep transistors in the DSTN structure.

Shown in Fig. 8 is the comparison between the estimation obtained by GA-based algorithm and simulation-based algorithm. Each point represents an cluster belonging to certain circuit in Table III, and x coordinate is the estimated $\mathrm{MSSC}_{\mathrm{clu}}$ obtained by simulation-based algorithm, and y is that of GA-based algorithm. As shown in the figure, most points are located in the area where simulation-based estimation is larger than that of GA-based algorithm, which matches (40).

Table III reports the running time for the two algorithms. Obviously, the simulation-based algorithm has a great advantage over the GA based algorithm in terms of speed. In general, it speeds up by two to three orders. Note that the speed of the simulation-based algorithm mainly depends on the number of the input vectors to build $\mathbf{S}$. This number is empirically decided and roughly linear to the size of the circuit. The efficiency of the simulation-based algorithm is very important for our sleep transistor sizing algorithm because otherwise it is very time-consuming to handle large size circuits, especially for CBSD, in which the procedure is evoked excessively to minimize $\sum_i \mathrm{MSSC}_{\mathrm{clu}_i}$.

REFERENCES

[1] J. Kao, S. Narendra, and A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2002, pp. 141–148.

[2] S. Mutah *et al.*, "1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos," *IEEE J. Solid-State Circuits*, pp. 847–854, Aug. 1995.

[3] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukuda, T. Kaneko, and J. Yamada, "A 1-v multithreshold-voltage CMOS digital signal processor for mobile phone application," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1795–1802, Nov., 1996.

[4] N. Shibata, H. Morimura, and M. Watanabe, "A 1-v, 10-mhz, 3.5-mw, 1-mb MTCMOS sram with charge-recycling input/output buffers," *IEEE J. Solid-State Circuits*, vol. 34, pp. 866–876, June, 1999.

[5] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," in *Proc. DAC*, 1998, pp. 495–500.

[6] M. Anis, S. Areibi, and M. Elmasry, "Dynamic and leakage power reduction in MTCMOS circuits using an automated efficient gate clustering technique," in *DAC*, 2002, pp. 480–485.

[7] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, pp. 584–594, Apr. 1990.

[8] S. Chowdhury and J. S. Barkatullah, "Estimation of maximum current in MOS IC logic circuts," *IEEE Computer-Aided Design*, vol. 9, pp. 642–654, June 1990.

[9] H. Kriplani, F. Najm, and I. Hajj, "Improved delay and current models for estimating maximum currents in CMOS VLSI circuits," in *Proc. IEEE Int. Symp. Circuits Systems*, 1994, pp. 435–438.

[10] Y. M. Jiang, A. Krstic, and K. T. Cheng, "Estimation for maximum instantaneous current through supply lines for CMOS circuits," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 61–73, Feb. 2000.

[11] Y. M. Jiang, K. T. Cheng, and A. C. Deng, "Estimation of maximum power supply noise for deep sub-micron designs," in *IEEE Proc. Symp. Low-Power Electronics and Design*, 1998, pp. 233–238.

[12] S. Kim, S. Kosonocky, and D. Knebel, "Understanding and minimizing ground bounce during mode transition of power gating structures," in *Int. Symp. Low Power Electronics and Design*, Aug. 2003, pp. 22–25.

[13] P. Heydari and M. Pedram, "Ground bounce in digital VLSI circuits," *IEEE Trans. VLSI Syst.*, vol. 11, pp. 180–193, Apr. 2003.

[14] X. D. Tan and C. J. Shi, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," in *Proc. Design Automation Conf.*, 1999, pp. 78–83.

[15] H. Su, J. Hu, S. S. Sapatnekar, and S. R. Nassif, "Congestion-driven codesign of power and signal networks," in *Proc. Design Automation Conf.*, 2002, pp. 64–69.

[16] S. X. D. Tan and C. J. Shi, "Efficient vlsi power/ground network sizing based on equivalent circuit modeling," *IEEE Computer-Aided Design*, vol. 22, pp. 277–284, Mar. 2003.

[17] S. Boyd, L. Vandenberghe, A. E. Gamal, and S. Yun, "Design of robust global power and ground networks," in *Proc. ISPD*, Apr. 2001, pp. 60–65.

[18] S. Chowdhury, "Optimum design of reliable IC power networks having general graph topologies," in *Proc. Design Automation Conf*, 1989, pp. 787–790.

[19] H. Kriplani, F. N. Najm, and I. N. Hajj, "Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: Algorithms, signal correlations, and their resolution," *IEEE Computer-Aided Design*, pp. 998–1012, Aug. 1995.

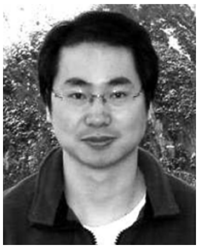
[20] A. Kristic and K. T. Cheng, "Vector generation for maximum instantaneous current through supply lines for CMOS circuits," in *Proc. Design Automation Conf.*, June 1997, pp. 383–388.

[21] Y. M. Jiang, K. T. Cheng, and A. Krstic, "Estimation of maximum power and instantaneous current using a genetic algorithm," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1997, pp. 135–138.

[22] E. M. Sentovich *et al.*, "SIS: A System for Sequential Circuit Synthesis,", Memorandum no. UCB/ERL M92/41, May 1992.

[23] W. Liao and L. He, "Leakage power modeling and reduction with data retention," in *Proc. Int. Conf. Computer Aided Design*, 2002, pp. 714–719.

[24] C. Long, J. Xiong, and L. He, "On optimal physical synthesis of sleep transistors," in *Proc. Int. Symp. Physical Design*, 2004, pp. 156–161.



Changbo Long (S'04) received the B.S.E.E. and M.S.E.E. degrees from Tsinghua University, Tsinghua, China, in 1999 and 2001, respectively, and the M.S. degree in computer engineering from the University of Wisconsin, Madison in 2003. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering at the University of California at Los Angeles (UCLA).

His research interests include computer-aided design of VLSI circuits and systems, power-efficient computing, and interconnect modeling and optimization.

Lei He (S'94–M'99) received the B.S. degree in electrical engineering from Fudan University, Shanghai, China, in 1990 and the Ph.D. degree in computer science from the University of California at Los Angeles, (UCLA) in 1999.

He is currently an Assistant Professor in the Department of Electrical Engineering at UCLA. From 1999 to 2001, he was a faculty member at University of Wisconsin, Madison. He has held industrial positions with Cadence, Hewlett-Packard, Intel and Synopsys as well. His research interests include computer-aided design of VLSI circuits and systems, interconnect modeling and design, programmable logic and interconnect, and power-efficient circuits and systems.

Dr. He received the Dimitris N. Chorafas Foundation Prize for Engineering and Technology in 1997, the Distinguished Ph.D. Award from the UCLA Henry Samueli School of Engineering and Applied Science in 2000, the NSF CAREER award in 2000, the UCLA Chancellor's Faculty Development Award in 2003, and the IBM Faculty Award in 2003.