# Distributed Sleep Transistor Network for Power Reduction*

Changbo Long
ECE Department
University of Wisconsin, Madison
clong@cae.wisc.edu

Lei He
EE Department
UCLA
lhe@ee.ucla.edu

## ABSTRACT

Sleep transistors are effective to reduce dynamic and leakage power. The cluster-based design was proposed to reduce the sleep transistor area by clustering gates to minimize the simultaneous switching current per cluster and then inserting a sleep transistor per cluster. In the 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.

## Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles.

## General Terms

Design.

## 1. 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's threshold voltage $V_t$ should be also reduced, which causes exponentially increase in the sub-threshold leakage current [1]. Multi-threshold CMOS (MTCMOS, see figure 1) has been introduced with low $V_t$ modules connected to ground through high $V_t$ transistors called sleep transistors [2]. The sleep transistor is turned off to reduce dynamic and leakage power in the standby mode, and is turned on to retain functionalities in the active mode.

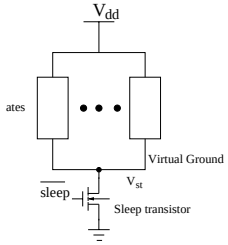


**Figure 1: MTCMOS circuit structure**

In this paper, we propose a novel distributed sleep transistor network (*DSTN*) with inherent advantages in area and performance compared to module-based and cluster-based sleep transistor designs [3, 4]. We will discuss background knowledge in Section 2, introduce the concept of DSTN in Section 3, and propose a gate-level DSTN synthesis methodology in Section 4. We will present experiments of gate-level synthesis and custom layout design in Section 5 and conclude in section 6. Proofs of all theorems can be found in the technical report[5].

## 2. BACKGROUND

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

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

where $C_L$ is the load capacitance, $V_{tL}$ is the threshold voltage in the low $V_t$ module, and $\alpha$ is the velocity saturation index for modeling short channel effects [6]. When the sleep transistor is present and the source drain voltage drop is $V_{st}$, the gate propagation delay increases to

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

In order to measure the increase in propagation delay, the following performance loss (PL) is defined [4]:

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

*This research is partially supported by NSF CAREER Award 0093273, SRC grant HJ-1008, and a grant from Intel. We used computers donated by Hewlett-Packard, Intel and SUN Microsystems. Address comments to lhe@ee.ucla.edu.


*DAC 2003,* June 2–6, 2003, Anaheim, California, USA.

According to the analysis in [4], for $PL = \delta$, we have

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

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

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

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 $V_{tL} = 350mV$ and $V_{tH} = 500mV$ in this paper), and $R_{st}$ is the channel resistance of the sleep transistor in the linear-operation region. The tolerable performance loss can be different among sleep transistors. For example, if all the gates in a particular low $V_t$ module are not in the critical path, large performance loss in this module is acceptable as long as the module does not become a part of the critical path. For the simplicity of presentation, we assume in this paper that all modules share the same PL value. On the other hand, $I_{st}$ is different with respect to different input vectors. To guarantee that the PL constraint holds for all possible input vectors, the maximum simultaneous switching current (MSSC) should be used to replace $I_{st}$ in (6) to calculate the size of the sleep transistor.

Our work is inspired by the following MSSC analysis. In cluster-based designs [4], one cluster of gates share one sleep transistor. The area of the sleep transistor $st_i$ is proportional to $MSSC(c_i)$, where $c_i$ denotes a cluster. If sleep transistors are ideally sized, the total area $A$ of all sleep transistors is proportional to $\sum_i MSSC(c_i)$, i.e., $A = k * \sum_i MSSC(c_i)$ with $k$ being a constant. On the other hand, the entire circuit is accommodated only by one large sleep transistor in module-based designs [3]. The ideal size of this sleep transistor is $k * MSSC(ckt)$, where $MSSC(ckt)$ is MSSC of the entire circuit. Note that $MSSC(ckt)$ is much smaller than $\sum_i MSSC(c_i)$ when the cluster size is small. For example, consider an extreme case that every cluster contains only one gate. $\sum_i MSSC(c_i)$ is the sum of the peak current for all gates. $MSSC(ckt)$ 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, $MSSC(ckt)$ is much smaller than $\sum_i MSSC(c_t)$. Therefore, the module-based design saves sleep transistor area compared to the cluster-based design.

Let virtual-ground wires be interconnects connecting the sleep transistor to low-$V_t$ gates. The above analysis does not consider the virtual-ground wires. The module-based design, however, leads to long virtual-ground wires as pointed out in [4]. The increased resistance of virtual-ground wires has to be compensated by more area in the sleep transistor. Such overhead can be avoided by having a local sleep transistor per cluster, and sleep transistor area can be further reduced by clustering gates to minimize the MSSC in the cluster. Minimizing MSSC introduces extra constraints for placement, and may conflict with timing-driven placement. In the next section, we will propose our DSTN design, and show that DSTN has a reduced area for both sleep transistors and virtual-ground wires, and is compatible with timing-driven placement. Owing to the fact that the cluster-based design is better than the module-based design [4], we compare DSTN mainly with cluster-based design in the rest of the paper.

# 3. SLEEP TRANSISTOR NETWORK

We illustrate the cluster-based sleep transistor design in figure 2.(a), where gates in a cluster are connected to the sleep transistor for this cluster by virtual-ground wires. Virtual-ground wires of different clusters are not connected. By adding more wires to form a mesh containing all virtual-ground wires, we obtain the DSTN structure in figure 2.(b). We assume that all sleep transistors share a common control signal in both designs.

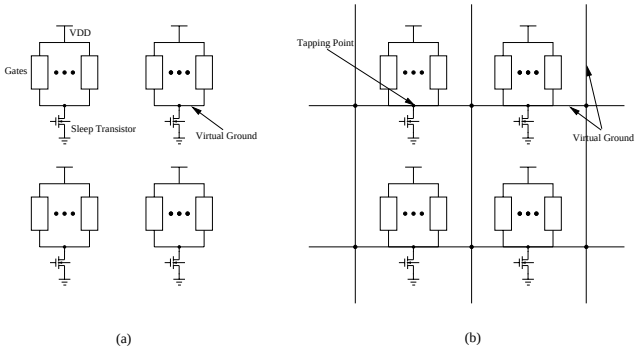


**Figure 2: (a) Cluster-based design, and (b) Distributed sleep transistor network**

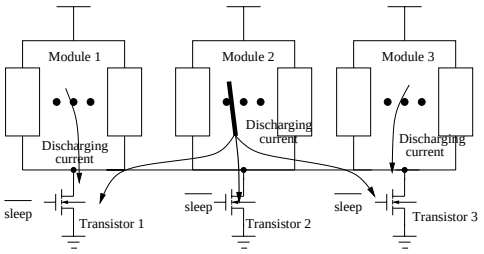


**Figure 3: Current discharging balance in DSTN**

We will show that DSTN reduces the sleep transistor area compared to the cluster-based design. The area saving can be explained by the discharging current balance phenomenon. As shown in figure 3, the switching current in module 2 is larger than those in module 1 and module 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 current from module 2 flowing to transistors 1 and 3[1]. The total area of all the sleep transistors in DSTN can thus be significantly reduced with presence of such current discharging balance. However, owing to the parasitic resistance and capacitance in virtual-ground wires, the total transistor area should be larger than the following

$$\frac{MSSC(ckt)}{\delta\mu_n C_{ox}(V_{dd} - V_{tL})(V_{dd} - V_{tH})}, \tag{7}$$

which is the optimum area for the single sleep transistor in the module-based design introduced in [3], and also the ideal total sleep transistor area in DSTN.

[1]A similar discharging current balance has been discussed in P/G modeling [7].

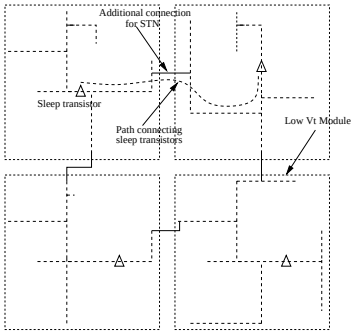


**Figure 4: Physical design of the STN**

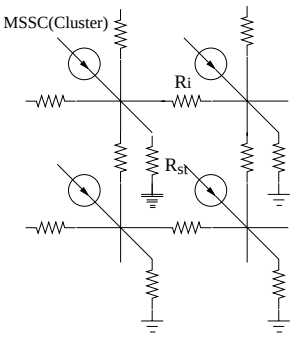


**Figure 5: Resistance network modeling of the DSTN**

The routing area overhead is a crucial aspect for all three types of sleep transistor design because every gate in the circuit has to be connected to a sleep transistor. Different sleep transistor designs impose different requirements for routing in terms of wire length and wire size. We assume in this paper that sleep transistors are connected to the ideal ground. Although DSTN and the module-based design may have the same topology for virtual-ground wires, the wire size for DSTN is found to be smaller due to the proximity of sleep transistors. On the other hand, DSTN needs more virtual-ground wire segments than the cluster-based design. As illustrated by the DSTN layout in figure 4, where the dotted lines are virtual-ground wires inside modules and are required by both DSTN and cluster-based design. Solid lines are virtual-ground wires that are needed by DSTN. These solid lines are short for compacted layout design. 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.

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

# 4. GATE LEVEL DSTN DESIGN

In this section, we first present the DSTN modeling, then formulate and solve the DSTN sizing problem. In order to compare different design styles, we will also introduce a rigorous algorithm for cluster-based sleep transistor design.

## 4.1 DSTN modeling

We model both sleep transistors and virtual-ground wires as resistors. Therefore, DSTN can be modeled as a resistance network shown in figure 5, with resistance $R_{st}$ for a transistor and $R_i$ for a virtual-ground interconnect. Note that $R_i$ is needed to accurately model the discharge current balance. Exact estimation of $R_i$, however, requires detailed layout information. In gate level design, we assume that $R_i$ is uniform for each wire. Specifically, we assume that the wire resistance is $0.05\Omega/\mu m$. We consider virtual-ground wires that are $200\mu m$ and $1000\mu m$ long, i.e., we consider $R_i = 10\Omega$ and $50\Omega$, respectively. Given our assumption that each cluster has about six gates (decided by the typical sleep transistor size in section 4.2), $200\mu m$ is a conservative estimation for virtual-ground wires between clusters, and $1000\mu m$ serves as the worst-case scenario to analyze the impact of $R_i$.

## 4.2 DSTN sizing

### 4.2.1 Problem formulation

We assume in this paper that the topology of DSTN is defined as *a priori*, and formulate the following DSTN sizing problem:

FORMULATION 1. **DSTN sizing problem(DSTN-SP):** *For given DSTN topology, the DSTN-SP problem finds a size for every sleep transistor in DSTN such that the total area of DSTN is minimized and the PL constraint is satisfied for every cluster.*

Note that DSTN sizing is totally different from the sleep-transistor sizing in the cluster-based design. The size of the sleep transistor in the cluster-based design is solely determined by the MSSC of the accommodated cluster. Owing to discharging current balance in DSTN, the size of a sleep transistor in DSTN depends on the current going through the accommodated cluster, the adjacent clusters, and even non-adjacent clusters. This makes the DSTN sizing problem much harder than the sizing problem of the cluster-based design. More precisely, DSTN can be modeled by a resistance network, and then the accurate transistor sizing can be obtained by algorithms similar to P/G sizing algorithms in [8]. We expect that well-designed heuristics may as well lead to good solutions, but in a more efficient fashion. We reveal below a few important properties in order to develop effective heuristics.

### 4.2.2 Properties

Note that our properties are based on an important observation about the resistance network: $R_i$ is normally much smaller than $R_{st}$. The channel resistance of the transistor in the linear-operation region is

$$R_{st} = \frac{1}{\mu_n C_{ox}(V_{dd} - V_{tH})}(\frac{L}{W}). \quad (8)$$

We assume $V_{tH} = 500mV$, a typical sleep transistor in DSTN has $\frac{W}{L} = 6$, and $V_{dd} = 1.3V$ in 100nm technology.

Thus, the typical resistance value for $R_{st}$ is around 218 Ω. On the other hand, a 200$\mu m$ long virtual-ground wire has $R_i$ of about 10Ω in 100nm technology. Therefore, it is reasonable to assume that $R_{st}$ is much larger then $R_i$.

THEOREM 1. *Assuming $R_i = 0$ and $PL = \delta$, the total transistor area in DSTN is determined by:*

$$\sum_i (\frac{W}{L})_i = \frac{MSSC(ckt)}{\delta \mu_n C_{ox}(V_{dd} - V_{tL})(V_{dd} - V_{tH})} \quad (9)$$

When $R_i = 0$, all sleep transistors in DSTN can be viewed as one single transistor with channel resistance and (W/L) of:

$$R = \frac{1}{\sum_i 1/R_{st_i}} \quad (10)$$

$$\frac{W}{L} = \sum_i (\frac{W}{L})_{st_i} \quad (11)$$

Because the current of the entire circuit goes through this single transistor, the following equation holds:

$$\frac{W}{L} = \frac{MSSC(ckt)}{\delta \mu_n C_{ox}(V_{dd} - V_{tL})(V_{dd} - V_{tH})} \quad (12)$$

Combining (12) and (11) leads to (9). We can also prove:

THEOREM 2. *To maintain PL as a constant, the total area of DSTN increases when $R_i$ increases.*

As $R_i$ increases, the effective resistance seen by the current source at each tapping point increases. Thus, the voltage drop in the sleep transistor increases when the current is constant. To maintain PL as a constant, the sleep transistor resistance has to be decreased, which results in more area consumption in DSTN.

The total area of DSTN can be roughly determined by Theorems 1 and 2 together. If $R_i = 0$, the total area of the DSTN is given by (9). However, according to Theorem 2, the total transistor area in DSTN must be larger than the value in (9). 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. *Assuming the current $I_i$ that flows into each tapping point $t_i$ being constant and the total area of the DSTN given, every transistor $st_i$ accommodating $t_i$ should be sized proportional to current $I_i$ in order to minimize the maximum voltage drop among all sleep transistors.*

Note that Theorem 3 is an ideal case to allocate area to individual transistors in DSTN. Although the current at each tapping point $t_i$ is not constant in real designs, Theorem 3 helps guiding the design of our DSTN transistor sizing scheme below.

#### 4.2.3 Algorithm

The overall flow of the sleep transistor sizing scheme is described as follows. We first calculate $MSSC(ckt)$ for example by genetic algorithm [9]. We then compute the total area in DSTN according to the following formula:

$$A = (1 + \beta)\frac{MSSC(ckt)}{\delta \mu_n C_{ox}(V_{dd} - V_{tL})(V_{dd} - V_{tH})}. \quad (13)$$

Our experiment shows that $\beta$ should range from 0.05 to 0.5 and a larger $\beta$ should be used for a bigger circuit. Finally, according to Theorem 3, the total DSTN area is allocated to each sleep transistor $st_i$ proportionally to the correspondent cluster MSSC.

### 4.3 Cluster based sleep transistor insertion

The total area of sleep transistors for the cluster-based sleep transistor design is proportional to $\sum_i MSSC(c_i)$. Clustering together gates with different switching times helps reduce $\sum_i MSSC(c_i)$, and in turn reduce the total area of sleep transistors.

The cluster based sleep transistor insertion problem is formulated as follows,

FORMULATION 2. **Cluster based sleep transistor insertion:** *Given a circuit and cluster size, partition gates into clusters such that the sum of MSSC for these clusters, and in turn the total area of all sleep transistors is minimized.*

A cluster-based design methodology has been proposed with placement constraints [4]. In this paper, we target at reaching the maximum potential of sleep transistor area reduction. Therefore we propose to apply simulated annealing (SA) for $\sum_i MSSC(c_i)$ minimization without placement constraints. In SA, each cluster is associated with a cost of MSSC. The cost for the entire circuit is the sum of costs for all clusters. The objective is to minimize the cost for the entire circuit. We take advantage of the freedom that a gate can be assigned to any cluster. Specifically, two gates are randomly picked from two clusters and exchanged in each move. We start SA from temperature of 100 and terminate at 0.1. The number of moves at a particular temperature is 200x of the number of clusters in the circuit. After these moves, the temperature is decreased by a factor of 0.9.

### 4.4 Cluster MSSC calculation

The primary objective of MSSC calculation is to search the input vector space to identify the maximum switching current value. The genetic algorithm(GA) based [9] and automatic test pattern generation(ATPG) based algorithm[10] have been developed for MSSC estimation. We employ GA algorithm to calculate the MSSC for the entire module in this paper. However, GA algorithm is inefficient to calculate MSSC for a large number of clusters. Therefore, we propose an efficient heuristic algorithm for cluster MSSC calculations in this section. The reader who is only interested in experiments may skip section 4.4.

MSSC estimation searches for the maximum current value considering both switching time and input vector. In order to simplify the problem, we first solve the MSSC estimation problem at a fixed time, that is, we first estimate $MSSC(c, t)$ based on a small number of random simulations. For example, we want to estimate the maximum current for the cluster of gates G1 to G7. We first simulate the cluster for a number of random input vectors. The switching activities at time $t$ for all gates in all simulations can be encoded in a table shown in figure 6, where 1 stands for switching and 0 means no switching. For example, row S1 (i.e., simulation S1) means that G1, G2 and G6 switch while G3, G4, G5 or G7 do not switch. Although G3, G4, G5 and G7 never switch simultaneously with G1, G2 and G6 at S1, they may switch simultaneously with G1, G2 and G6 under

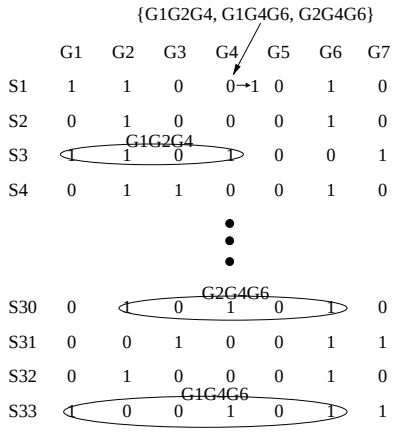


**Figure 6: Database containing pre-simulations for cluster MSSC**

other input vectors. In this case, the switching current at those input vectors is larger than the one in S1. We want to capture this potential and expand the list of simultaneous switching gates as much as we can. We illustrate the idea of expanding simultaneous switching lists by using list S1, in which the simultaneous switching gates are G1, G2 and G6. Instead of checking whether G4 can switch simultaneously with G1, G2 and G6, we check whether all the combinations of three gates, i.e., G1G2G4, G1G4G6 and G2G4G6 can switch simultaneously. As shown in figure 6, G1G2G4, G2G4G6 and G1G4G6 do happen in S3, S30 and S33. Thus, G4 has a large potential to switch simultaneously with G1, G2 and G6. G4 is then set to be switching at S1 and the switching current of G4 is added into the total switching current of S1. The switching list for each simulation is expanded until no more expansion is possible. The maximum current value among all the simulations is $MSSC(c,t)$.

Overall, our method for cluster MSSC estimation contains two phases. In the first phase, we carry out a number of random simulations and choose the time (called peak time) for the peak current of each simulation. In the second phase, we apply the above $MSSC(c,t)$ for every peak time.

# 5. EXPERIMENT RESULTS

## 5.1 Gate level synthesis

All proposed algorithms have been implemented inside SIS[11] environment. We use ISCAS benchmark circuits and report experiment results in Table 1. 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 sub-section.

We first compare the area (i.e., transistor width) used by DSTN and cluster-based design (CB-STD), respectively. We measure area by the total channel width of sleep transistors. One can see that DSTN uses significantly smaller area than CB-STD does. On average, the area reduction is 49.8%. Because we do not consider the delay constraint during placement for CB-STD, we obtain a lower bound of the cluster MSSC in a timing-driven placement and in turn a lower bound of the sleep transistor area in CB-STD. Therefore, the area reduction by DSTN would be larger compared to CB-STD if considering practical placement constraints.

We then compare performance loss. We have used extensive random simulations to verify the quality for both sizing schemes. Specifically, 10,000 random simulations for each circuit have been conducted to calculate the maximum PL (in short, MPL) for each circuit. For DSTN, the peak current for each module in each simulation is applied to the resistance network as the current source. We compute the transistor channel resistance by (8), and use $R_i = 10\Omega$ and $50\Omega$ for virtual-ground wires. The resulting resistance network is solved by a sparse linear equation solver integrated with SIS. The calculated voltages at tapping points are used to compute the performance loss via (3). Note that the resulting MPL value in Table 1 is an upper bound of MPL for the following reasons: (i) the above $R_i$ values are conservative as discussed in Section 4.1. (ii) the peak current for an individual module normally happens at different times, but we assume that all the peak current happens at the same time in our experiment.

The same random simulations have been applied to calculate MPL in CB-STD, where PL is calculated via (6). Although the peak current value is also used to calculate MPL, it will not overestimate the PL because each module only discharge from one sleep transistor in CB-STD. Instead, ignoring the resistance of virtual ground in CB-STD leads to a lower bound estimation.

As shown in Table 1, when $R_i = 10\Omega$ (a conservative case as discussed in Section 4.1), MPL of DSTN is on average 10% smaller than that of CB-STD. When $R_i = 50\Omega$, i.e., an extreme worst-case scenario as discussed in Section 4.1, MPL of DSTN is about 6% worse than CB-STD. However, $R_i$ is normally $5 - 15\ \Omega$ in 100nm technology when the cluster size is 6. Furthermore, the MPL presented in Table 1 is an upper bound of the real MPL in DSTN, and is a lower bound of the real MPL in CB-STD. Therefore, it is fair to say that DSTN is better than CB-STD in terms of MPL.

Note that MPL for both DSTN and CB-STD are larger than 5%, the PL bound in our experiments. It is because the current values in a large number of random simulations may be bigger than the estimated cluster MSSC. This underdesign can be easily removed by scaling up the estimated MSSC.

## 5.2 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, cluster-based sleep transistor design(CB-STD) 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, is implemented. Then, a CB-STD 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

**Table 1: Area and MPL for DSTN and CB-STD**

| Circuit | #Gate* | #PI | #PO | Area (W/L) | | | MPL(%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | CB-STD | DSTN | Reduction (%) | CB-STD (lower bound) | DSTN (upper bound) | |
| | | | | | | | | $R_i = 10$ | $R_i = 50$ |
| C432 | 323 | 36 | 7 | 439 | 205 | 53.3 | 7.04 | 6.38 | 7.25 |
| C499 | 640 | 41 | 32 | 929 | 533 | 42.6 | 7.69 | 5.33 | 6.20 |
| C880 | 528 | 60 | 26 | 801 | 581 | 27.5 | 6.25 | 4.07 | 4.88 |
| C1355 | 625 | 41 | 32 | 878 | 532 | 39.4 | 7.53 | 4.36 | 5.73 |
| C1908 | 830 | 33 | 25 | 1286 | 416 | 67.7 | 7.01 | 8.54 | 9.60 |
| C2670 | 1459 | 233 | 140 | 1951 | 789 | 59.6 | 6.87 | 7.03 | 8.15 |
| C3540 | 1613 | 50 | 22 | 2715 | 796 | 70.7 | 8.95 | 8.72 | 10.06 |
| C5315 | 2813 | 178 | 123 | 4659 | 2302 | 50.6 | 7.62 | 5.25 | 6.47 |
| C6288 | 2464 | 32 | 32 | 6219 | 3640 | 41.5 | 5.18 | 6.61 | 7.85 |
| C7552 | 3685 | 207 | 108 | 6156 | 3377 | 45.1 | 5.16 | 6.41 | 7.29 |
| Avg. | – | – | – | – | – | 49.8 | 6.93 | 6.27 | 7.35 |

*Circuit gate number after mapping in SIS.

together and have a same size[2]. As in the cluster-based design, sizes of the sleep transistors in DSTN are determined by SPICE simulations to make PL below 5%.

| Properties | ST-free | CB-STD | DSTN |
|---|---|---|---|
| Leakage($nA$) | 59.80 | 5.72 | 1.23 |
| Critical path delay($nS$) | 1.66 | 1.79 | 1.68 |
| ST area($\mu m^2$) | 0 | 1449.6 | 212.2 |
| Chip area($\mu m^2$) | 11960.0 | 13892.0 | 12880.0 |

**Table 2: Layout design comparison.**

As shown in Table 2, compared to the ST-free design, both CB-STD and DSTN achieve significant leakage current reduction but DSTN is approximately five times better than CB-STD. Both CB-STD and DSTN increase the critical path delay but DSTN has a much smaller delay than CB-STD. DSTN has a transistor area several times smaller than CB-STD. These comparisons are consistent with previous theoretical analysis and experiment results.

## 6. CONCLUSION AND FUTURE WORK

Sleep transistors are effective to reduce both dynamic and leakage power. 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 lost. Having these advantages, DSTN can be used to implement power gating for reducing dynamic and leakage power [12].

Sleep transistor can be viewed as an essential part of the power/ground network. We assume that the power/ground network (both global and virtual) is given *a priori* in this study, and plan to investigate the co-design of DSTN and power/ground network in the future.

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

## 7. REFERENCES


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

[2] S.Mutah and et al., "1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos," *IEEE Journal of Solid-State Circuits*, August 1995.

[3] J. Kao, S. Narendra, and A. Chandrakasan, "Mtcmos hierarchical sizing based on mutual exclusive discharge patterns," in *DAC*, 1998.

[4] 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.

[5] C. Long and L. He, "Distributed sleep transistor network for power reduction," tech. rep., UCLA EE Dept., http://eda.ee.ucla.edu/publications.html, 2003.

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

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

[8] 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*, pp. 78–83, 1999.

[9] 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..*, pp. 135–138, May 1997.

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

[11] E. M. Sentovich, K. J. Singh, L. Lavagno, and etc., "Sis: a system for sequential circuit synthesis," *Memorandum NO. UCB/ERL M92/41*, May 1992.

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