Sleep Transistor Distribution in Row-Based MTCMOS Designs

Chanseok Hwang[1], Peng Rong[2], Massoud Pedram[3]
[1] Samsung Electronics, Seoul, South Korea
[2] LSI Logic Corp, Milpitas, CA, USA
[3] University of Southern California, Los Angeles, CA, USA

Abstract *- The Multi-Threshold CMOS (MTCMOS) technology has become a popular technique for standby power reduction. This technology utilizes high-Vth sleep transistors to reduce subthreshold leakage currents during the standby mode of CMOS VLSI Circuits. The performance of MTCMOS circuits strongly depends on the size of the sleep transistors and the parasitics on the virtual ground network. Given a placed net list of a row-based MTCMOS design and the number of sleep transistor cells on each standard cell row, this paper introduces an optimal algorithm for linearly placing the allocated sleep transistors on each standard cell row so as to minimize the performance degradation of the MTCMOS circuit, which is in part due to unwanted voltage drops on its virtual ground network. Experimental results show that, compared to existing methods of placing the sleep transistors on cell rows, the proposed technique results in up to 11% reduction in the critical path delay of the circuit.*

1. INTRODUCTION

Leakage power in modern CMOS VLSI circuits has become a component comparable to dynamic power dissipation. Typically, the subthreshold leakage current dominates the device off-state leakage due to low V_{th} transistors employed in logic cell blocks in order to maintain the circuit switching speed in spite of decreasing V_{DD} levels [1]. The Multi-Threshold CMOS (MTCMOS) technique can significantly reduce the subthreshold leakage currents during the circuit sleep (standby) mode by adding high-V_{th} power switches (sleep transistors) to low-V_{th} logic cell blocks [2][3]. This is because the stacked high-V_{th} sleep transistor connected to the bottom of the pull-down network of all logic cells in the circuit acts as a high-resistance element during the sleep mode, which limits the leakage current from V_{dd} to ground lines. At the same time, because of the stack effect, the subthreshold leakage of the low-V_{th} transistors in the logic block itself goes down. This leakage reduction is preferably achieved with small performance degradation because, during the active mode of the circuit, the sleep transistor is fully on (i.e., it operates in the linear mode), and thus, all low-V_{th} logic cells in the MTCMOS logic block can switch very fast.

GLSVLSI'07, March 11–13, 2007, Stresa-Lago Maggiore, Italy.

Unfortunately, the situation is different in real designs. More precisely, during the active mode of the circuit operation, the high-V_{th} sleep transistor acts as a small linear resistance placed at the bottom of the transistor stack to ground, causing the propagation delay of the cells in the logic block to increase. In addition, the virtual ground network itself acts as a distributed RC network, which causes the voltage of the virtual ground node to rise even further, thereby degrading the switching speed of the logic cells even more (cf. Fig.1.) The former effect is a function of the size of the sleep transistor whereas the latter effect is a function of the physical distance of the logic cell from the sleep transistor.

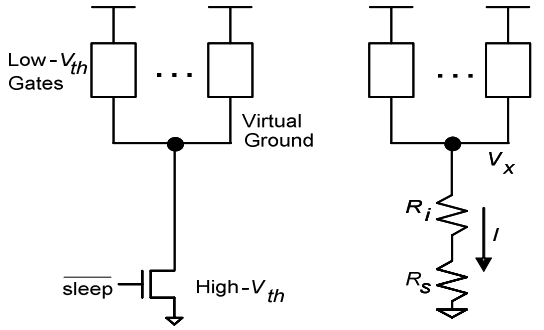


Figure 1. (a) MTCMOS circuit structure, (b) The circuit model with virtual ground interconnect and sleep transistor modeled as resistors, R_i and R_s, respectively

Figure 1(a) depicts a logic block, *LB*, in which a group of low-V_{th} logic cells are first connected to the virtual ground node and then through a high-V_{th} sleep transistor, *S*, to the actual ground, GND [2]. Figure 1(b) models the virtual ground interconnection and the high-V_{th} sleep transistor, which behaves like a linear resistor in the active mode of the circuit operation [4], as resistors R_i and R_s, respectively. The virtual ground is at voltage V_x above the actual ground, i.e., $V_x = I \cdot (R_s + R_i)$ where I is the current flowing through the virtual ground sub-network and the sleep transistor. The voltage drop across R_s + R_i reduces the gate over-drive voltage of MTCMOS logic cells (i.e., their V_{gs} value) from V_{dd} to $V_{dd} - V_x$.

In this paper, we present an optimal algorithm for placing sleep transistors for the standard cell-based layout design, which minimizes the performance degradation of MTCMOS circuits due to the interconnect resistance of the virtual ground network. We discuss previous works in section 2, introduce the problem formulation and the proposed method

in section 3 and 4, respectively. Then, we present experiment results in section 5 and conclude in section 6.

2. PRIOR WORK

Optimal sizing of the sleep transistors (STs) has been actively researched since the MTCMOS technology was introduced. One of the most conservative approaches is the dedication of one ST to each logic cell and the optimization of individual STs, which is known as "fine-grained" leakage control [5][6]. This approach makes it easier to do RT-level sign off by using standard static timing analysis techniques whereas it tends to incur large area overhead [7]. Other approaches [8][9] group logic gates based on their current discharge patterns in a given switching cycle so that sleep transistors of gates with mutually exclusive current discharge patterns are merged together, thereby, reducing the area overhead. In [8], the authors first size the sleep transistor of each logic cell so as to impose an upper bound on the performance degradation of the cell during the active mode of the circuit operation. Next, they calculate the discharge current patterns of all logic cells in the circuit based on a unit delay model, and finally, merge the sleep transistors of cells with non-overlapping discharge current waveforms. In [9], the authors use a more precise delay model to do the same steps, presenting several heuristic techniques for efficient gate clustering. In [10], the authors use average current consumption values of logic gates to determine the sleep transistor width needed to satisfy a required circuit speed based on the assumption that the circuit speed depends only weakly on the circuit operating pattern for sufficiently large sleep transistor sizes. In other approaches [11][12], the authors selectively apply the MTCMOS technology for gates belonging to timing-critical paths of a circuit, where low-V_{th} gates are only used in the timing-critical paths. Recently, a number of approaches have begun to consider the interconnect resistance of the virtual ground line as one of the crucial factors that affect the performance of MTCMOS circuits. In [13], the authors model both sleep transistors and virtual ground interconnects as resistors, and thereby, consider a resistive-only network when computing sizes of the distributed sleep transistors. In [14][15], the authors show that the delay and output slew of a gate with the MTCMOS technology increase linearly with virtual ground length, and then take this parameter into account when modeling the delay of MTCMOS gates.

This paper addresses the automatic placement of sleep transistors considering the interconnect resistance of the virtual ground in standard cell-based layout design. In [7][16], the authors provide design methodologies that treat a ST as a standalone library cell and then calculate and allocate a number of sleep transistor cells (STCs) on each cell row. The STCs are then placed at one or the other corner of each row on a row-by-row basis. In these methods, all cells in a row share all of the STCs and the virtual ground line in the same row. The main advantages of these approaches are that (a) they are fully compatible with the existing standard-cell physical design flow and (2) they result in a shorter re-activation time when exiting the sleep state [16]. However, these approaches do not consider the voltage drop due to the virtual ground interconnect in deciding the location of STCs. In practice, they tend to over-estimate the size of sleep transistors needed on each cell row so as to compensate for the voltage drop due to virtual ground interconnect resistance. With continued process scaling, the wire segments in the ground network become more resistive, causing more area overhead. In addition, the performance degradation of a logic cell that is far away from the STCs may greatly affect the overall circuit performance if the logic cell happens to lie on a timing-critical path of the circuit. Therefore, in this paper, we shall address the question of how to place STCs for the standard cell-based layout design so that the performance degradation of MTCMOS circuits due to the interconnect resistance of the virtual ground line is minimized.

3. PROBLEM FORMULATION

The problem of sleep transistor distribution can be described as follows. Given a placement of MTCMOS design in the row-based layout and the distribution of sleep transistor cells on each row, the objective is to determine an optimal placement of sleep transistor cells such that the performance degradation of the MTCMOS circuit due to the voltage drop of the virtual ground line is minimized.

Let's focus on the sleep transistor distribution in row *y*. Let $C=\{c_i, i=1,2,\ldots,n\}$ denote the set of logic cells in row *y*, and $S=\{s_j, j=1,2,\ldots,m\}$ denote the set of STCs to be placed on the row. *n* and *m* denote the cell and STC counts, respectively. We define the performance loss of a logic cell $c_i \in C$ in the active mode as follows:

$$PL(c_i) = \Delta T(c_i) - SL(c_i) \tag{1}$$

where $\Delta T(c_i)$ denotes the additional propagation delay of cell c_i that results from the resistance of the virtual ground. $SL(c_i)$ is the slack available for cell c_i before inserting the STCs. The slack times are calculated by doing static timing analysis (STA) on the circuit.

Using the well-known alpha-power delay model [17], the cell propagation delay in the presence of a sleep transistor and virtual ground interconnects is given by

$$T(c_i) \propto \frac{C_{load,i} \cdot V_{dd}}{(V_{dd} - I_{STC} \cdot (R_{STC} + R_i) - V_{tL})^{\alpha}} \tag{2}$$

where $C_{load,i}$ and V_{tL} denote the capacitive loading seen by cell I and the threshold voltage of the low V_{th} transistors in the cell, respectively. R_{STC} denotes the STC resistance, R_i denotes the interconnect resistance, and $I_{STC}(R_{STC}+R_i)$ is the source voltage at the bottom of the NMOS-section of the logic cell. Thus, we may calculate $\Delta T(c_i)$ as

$$\Delta T(c_i) = T(c_i) - T(c_i)|_{R_i=0} \tag{3}$$

R_i is dependent on the distances between c_i and STCs on this cell row, $s_j \in S$. Accordingly, $\Delta T(c_i)$ is a function *f* of these distances:

$$\Delta T(c_i) = f(|x_{s_1} - x_{c_i}|, |x_{s_2} - x_{c_i}|, \ldots, |x_{s_m} - x_{c_i}|) \tag{4}$$

where x_{ci} and x_{sj} denote the horizontal coordinates of logic cell c_i and STC s_j, respectively. Furthermore, to a first order, $\Delta T(c_i)$ is dominated by the position of the sleep transistor that

is closest to c_i. Let d_i denote the minimal distance between c_i and any of the sleep transistors, i.e.,

$$d_i = \min(|x_{s_1} - x_{c_i}|, |x_{s_2} - x_{c_i}|, ..., |x_{s_m} - x_{c_i}|) \quad (5)$$

We may approximate equation (4) by

$$\Delta T(c_i) \simeq g(d_i) \quad (6)$$

Now, in equation (2), R_i may be replaced with $r \cdot d_i$, where r is the wire resistance per unit length of the virtual ground network. By using Taylor Series Expansion of the right-hand side of equation (2), we can obtain function g in equation (6):

$$\Delta T(c_i) = \frac{\partial T(c_i)}{\partial R_i} \frac{\partial R_i}{\partial d_i}\bigg|_{d_i=0} \cdot d_i \quad (7)$$

That is,

$$\Delta T(c_i) = w_i \cdot d_i$$

$$\text{where } w_i = \kappa_i \frac{\alpha C_{load,i} \cdot I_{STC} \cdot r \cdot V_{dd}}{(V_{dd} - I_{STC} \cdot R_s - V_{tL})^{\alpha+1}} \quad (8)$$

κ_i is a proportionality coefficient

Thus the performance loss can be written as

$$PL(c_i) = w_i \cdot d_i - SL(c_i) \quad (9)$$

We next define a cost figure, Φ, for the STC placement:

$$\Phi = \max_{1 \le i \le n} PL(c_i) \quad (10)$$

It is important to balance the current flowing through each sleep transistor in the active mode. Otherwise, a sleep transistor that carries too much current will significantly increase the virtual ground voltage in its physical neighborhood, hence, it will decrease the performance of the cells in the region. Thus, the optimization problem for *STC distribution* may be formulated as

$$\text{Minimize } \Phi \quad (11)$$

subject to

$$I_{s_j} \le I_{s_{\max}}, \quad j = 1, 2, ..., m \quad (12)$$

where I_{sj} denotes the total current passing through STC s_j and I_{smax} is an upper bound on the current flowing through the STC. (The rationale is that when it was decided that cell row R should have m sleep transistors on it, that decision was in part based on the assumption that no sleep transistor will have to carry more than I_{smax}.) The minimum Φ value will be denoted by $\Phi_{\min}$.

4. PROPOSED APPROACH

In this section, we present an optimal solution to the STC distribution problem where $m<n$. For $m \ge n$, it is easy to see that $\Phi_{\min}$ can be achieved trivially by enforcing $x_{si}=x_{ci}$, $i=1,2,\ldots,n$. Let $P=\{p_j, j=1,2,\ldots,m\}$ denote a partition of the set of cells C in row R. Let x_{pj} denote coordinate of the rightmost cell in p_j.

Definition 1: A *valid* partition is a partition that satisfies the following two constraints: 1) $x_{p1}<x_{p2}<\ldots<x_{pm}$; 2) any cell c_i belonging to section p_j satisfies the condition that $x_{pj-1}<x_{ci} \le x_{pj}$. All valid partitions compose the valid partition space, Ω.

Definition 2: A *true* solution to the STC placement is a valid partition that additionally satisfies the following: 1) there is exactly one sleep transistor s_j in each section p_j; 2) within each section

$$\Phi_{pj} \triangleq \max_{c_i \in p_j} \{g(|x_{s_j} - x_{c_i}|) - SL(c_i)\} \quad (13)$$

is minimum. All true solutions compose true solution space, Σ.

***Lemma 1*:** Consider a true solution σ on the valid partition $P=\{p_j, j=1,2,\ldots,m\}$. Define $\Psi = \max_{1 \le j \le m} \Phi_{pj}$, where Φ_{pj} is defined in equation (13). We have: $\Phi \le \Psi$.

Proof: Assume $\Phi = g(|x_{s_j} - x_{c_i}|) - SL(c_i)$, where c_i is a cell and s_j is the sleep transistor associated with section p_j. There are two cases to consider. First, if $c_i \in p_j$, then $\Phi \le \Phi_{p,j} \le \Psi$, and the proof is complete. Second, if $c_i \notin p_j$, assume $c_i \in p_k$ (with sleep transistor s_k), then from equation (5):

$$\Phi \le g(|x_{s_k} - x_{c_i}|) - SL(c_i) \le \Phi_{pk} \le \Psi. \square$$

Lemma 2: Let Σ_m denote the true solution space for m sleep transistors and Ψ^*_m denote the minimal Ψ value on space Σ_m. The following monotone property holds: $\forall m_1<m_2$, there exists $\Psi^*_{m2} \le \Psi^*_{m1}$.

Proof: Given $m<n$, assume $\sigma_m \in \Sigma_m$ is a true solution on a valid partition P. There must exist a section p_j that contains more than one cell. The corresponding sleep transistor is denoted by s_j. Divide p_j into two new sections p'_j and p'_{j+1}, where p'_{j+1} contains only cell c_i, the rightmost one in p_j, i.e., $x_{ci}=x_{pj}$. Obviously, the new partition P' after this division is also a valid partition. Now add a new sleep transistor for partition P' and assign it to location x_{ci}. Thus $\Phi_{p'j+1} = -SL(c_i)$, which is less than Φ_{pj} based on definition (13). A new true solution $\sigma_{m+1} \in \Sigma_{m+1}$ on the valid partition P' can be obtained by shifting s_j so that $\Phi_{p'j}$ is minimized while keeping other sleep transistors unaffected. From Definition 2, there exists $\Phi_{p'j} \le \Phi_{pj}$, which leads to $\Psi_{\sigma m+1} \le \Psi_{\sigma m}$. Since σ_m is arbitrarily selected, the proof is complete. $\square$

***Theorem 1*:** The solution with a minimal Ψ value in space Σ is an optimal solution to the STC distribution problem defined by equations (11) and (12).

Proof: Let σ^* denote the solution in space Σ_m that has the minimal Ψ value. Assume σ_{opt} is an optimal solution to the problem defined by equations (11) and (12). Based on σ_{opt}, we can construct a valid partition P of set C as follows. First, we sort sleep transistors in an increasing order of their coordinates. Next, each cell is assigned to the section corresponding to its closest sleep transistor. Let m' denote the number of sections in P, where $m' \le m$. Notice that m' may be less than m because it is possible that a sleep transistor is not the closest one to any cell. Similar to the proof of Lemma 1, we can easily show that $\Phi_{\sigma_{opt}} \le \Psi_{\sigma_{opt}}$. Conversely, from the construction procedure for partition P, we know

$\forall c_i \in p_j$, $PL(c_i) = g(|x_{s_j} - x_{c_i}|) - SL(c_i)$, thus $\Phi_{pj} = \max_{c_i \in p_j} PL(c_i) \le \Phi_{\sigma_{opt}}$, which is followed by $\Psi_{\sigma_{opt}} = \max_{1 \le j \le m'} \Phi_{p,j} \le \Phi_{\sigma_{opt}}$. So we have $\Phi_{\sigma_{opt}} = \Psi_{\sigma_{opt}}$. Since P is a valid partition for m' sleep transistors, according to Lemmas 1 and 2 and because $m' \le m$, it follows that $\Phi_{\sigma^*} \le \Psi_{\sigma^*} = \Psi^*_m \le \Psi^*_{m'} \le \Psi_{\sigma_{opt}} = \Phi_{\sigma_{opt}}$. So σ^* must also be an optimal solution to the STC distribution optimization problem and $\Phi_{\min} = \Psi_{\sigma^*}$. □

Based on Theorem 1, a simple approach for solving the STC distribution problem is to examine all possible valid partitions and select the true solution with the minimal Ψ value. However, for n cells and m sleep transistors in a row, the number of valid partitions is C_{m-1}^{n-1}. In the case that $m=n/2$, $C_{m-1}^{n-1} \approx \frac{2^n}{\sqrt{2\pi}}$. So if n and m are large, the brute-force approach will be impractical.

We present an efficient dynamic programming approach to search for the optimal solution. Before describing the approach, we first re-formulate the current balance constraint on STCs to facilitate its incorporation into the proposed approach. By assuming a uniform current distribution in the row, the maximal current constraint can be described as a limit on the number of cells that any section in a valid partition can include. Let pn_j denote the number of cells in section j and $pn_{\max}$ denote the preset upper bound of pn_j. Constraint (12) can be restated as

$$\begin{aligned} & pn_j \le pn_{\max}, \quad j = 1, 2, ..., m \\ & pn_{\max} \ge n/m \end{aligned} \tag{14}$$

Assume the cell set C in row R is already sorted from left to right. Define function $\Gamma(i, j, k)$ to represent the minimal Φ value for k sleep transistors on a subset of cells in C from c_i to c_j, where $1 \le i \le j \le n$, $1 \le k \le m$. Thus $\Phi_{\min} = \Gamma(1, n, m)$. From Theorem 1, we have

$$\Gamma(1, j, k) = \min_{\substack{\max(j-(k-1)\cdot pn_{\max},1) \le l \\ l \le \min(pn_{\max}, j)}} \max(\Gamma(1, j-l, k-1), \Gamma(j-l+1, j, 1)) \tag{15}$$

and

$$\Gamma(i, j, 1) = \min_{x_s} \max_{i \le l \le j} g_l(|x_s - x_{c_l}|) - SL(c_l) \tag{16}$$

In equation (15), l denotes the number of cells in the rightmost section between c_1 and c_j. According to constraint (14), l should be no greater than $pn_{\max}$; on the other hand, l has to be large enough so that the remaining cells can be put into the remaining $k-1$ sections.

The pseudo-code for the optimal STC distribution algorithm is presented in Figure 2. First, Equation (16) is solved for all $\Gamma(i, j, 1)$ values, $1 \le i \le j \le n$, which are stored in a table with a dimension of $n \times n$. The proposed dynamic programming approach utilizes a table $\Gamma(j)$, $1 \le j \le n$, where each entry holds the computed value of $\Gamma(1, j, k)$ during the k_{th} iteration; and table $I(j, k)$, where each entry stores the l value in equation (15) which leads to the minimal $\Gamma(1, j, k)$. Lines 3 to 9 embody the dynamic programming iterations of updating values in tables $\Gamma(j)$ and $I(j, k)$. The decrement of j from n to i ensures that the updated entries of table $\Gamma(j)$ will not be used in the same k iterations, and thus, enables an in-place operation. The procedure in Line 11 traces back through pointers held in table $I(j, k)$ and works as follows. Read the value in the $I(j, k)$ entry. Assume $I(j, k)=l$, then assign cells from c_{i-l+1} to c_j to section p_k. Next, go to entry $I(j-l, k-1)$ and section p_{k-1}. This process continues until $k=1$. Finally, after partition P is constructed, the coordinates of sleep transistors are obtained by solving Equation (16) over each corresponding section in P.

STC Distribution Algorithm

INPUT: x_{ci}, $i=1,2,\ldots,n$, coordinates of n cells in a row R; m, number of sleep transistor cells to be placed in R

OUTPUT: $\Phi_{\min}$ and x_{sj}, $j=1,2,\ldots,m$, coordinates of m sleep transistor cells

1. $\Gamma(i,j,1) \leftarrow$ solution to equation (16), $\forall i,j$, $1 \le i \le j \le n$
2. $\Gamma(j) \leftarrow \Gamma(1, j, 1)$, $\forall j$, $1 \le j \le n$
3. for $k = 2:m$
4. for $j=n:1$ // j decrement
5. $t \leftarrow \infty$
6. for $l = \max(j-(k-1)pn_{\max},1):\min(pn_{\max}, j-k)$
7. $t \leftarrow \min(t, \max(\Gamma(i, j-l), \Gamma_1(j-l+1, j)))$
8. $I(k, j) \leftarrow l$
9. $\Gamma(j) \leftarrow t$
10. $\Phi_{\min} = \Gamma(n)$
11. Trace back from $I(m,n)$, and construct partition P of cells
12. $x_{sj} \leftarrow$ solution to equation (16) for each section p_j in partition P, $j=1,2,\ldots,m$

Figure 2. Optimal STC distribution algorithm in a row

It takes only $O(n \cdot pn_{\max})$ iterations to calculate table $\Gamma(i, j, 1)$; and in each iteration equation (16) is solved once, which in turn requires $O(pn_{\max})$ multiplications and additions. Dynamic programming embodied between line 3 to 9 takes $O(nm \cdot pn_{\max})$ iterations to obtain $\Phi_{\min}$, and finally the trace-back procedure in line 11 takes m steps. Thus, the total timing complexity is $O(nm \cdot pn_{\max} + n \cdot pn_{\max}^2)$, since $m<n$. Similarly, table $\Gamma(1, j, k)$ occupies $O(n \cdot pn_{\max})$ memory space, and table $I(j, k)$ $O(nm)$. Thus the spatial complexity of this algorithm is $O(n \cdot pn_{\max} + nm)$.

5. EXPERIMENTAL RESULTS

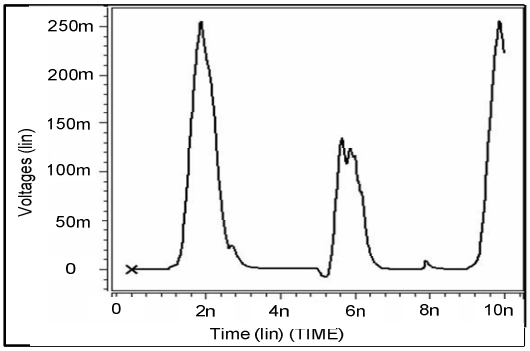
We used ISCAS benchmark circuits and SIS to generate optimized gate level netlists; all benchmarks were first optimized by using the SIS "script.rugged" and doing timing-driven technology mapping based on an industrial-strength 90nm ASIC design library. All benchmarks were run on SUN Ultra Spark II machine. We set the high V_{th} values for PMOS

and NMOS as –303mV and 260mV, and the low V_{th} values for PMOS and NMOS as –250mV and 200mV, respectively.

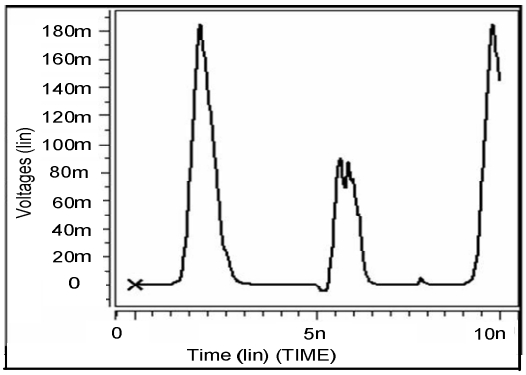
In our experiments, we first generated detailed row-based cell placements for the generated benchmark circuits by using the timing-driven placer of [18]. We then calculated the number of sleep transistor cells to be placed at each row based on the Average Current Method (ACM) of [10], where the size of every sleep transistor cell is determined so that its "on" resistance, R_s, comes out to be about 300Ω. Next we applied the proposed *Optimal* ST distribution technique (from here we call it **OSTD**) for each placement row, which is followed by a layout adjustment to remove overlaps between logic cells and sleep transistor cells.

We compare OSTD with the other sleep transistor cell placement methods used in [9], where all sleep transistor cells are located at the corners of each row so that they do not disturb the existing placement (from here we call this method CSTD for *Corner-based* ST distribution), in terms of the total wire length and critical path delay. In addition, we implement another sleep transistor cell placement method, which distributes the sleep transistor cells uniformly on the row, thereby, spacing equally between sleep transistors on the same row (from here we call this method USTD for *Uniform* ST distribution.) We also compare OSTD with USTD in terms of the critical path delay. To obtain these results, we perform global routing after the placement of sleep transistor cells so that the wiring loads of all nets may be accurately calculated. Next, we extract cell locations and interconnect parasitics and input the whole extracted netlist to HSPICE so that we measure the critical path delay for the benchmark circuits. To minimize the simulation time, we run STA before HSPICE simulation to identify the set of PI-PO critical paths and then apply input vectors that cause the propagation of an event along these paths. We therefore run HSPICE simulation only for the input vectors producing the transitions along the STA-identified timing-critical paths.

Figure 3 presents transient simulations of the virtual ground line of the first row in the layout of circuit C7552, operating at a supply voltage of 1V. Compared to CSTD, OSTD reduces the virtual ground bounce by about 50%, from 250mV to 180mV. This reduction of virtual ground bounce tends to improve the performance of the circuit.



(a) CSTD



(b) OSTD

Figure 3. Simulation results for virtual ground bounce

Table 1 shows comparison results between OSTD and CSTD in terms of critical path delay and wire length. For each circuit benchmark, we generated 10 different placement solutions (corresponding to different random seeds for the timing-driven placer.) Therefore, we also generated 10 different sets of wire lengths and critical path delays, and reported in Table 1 only the mean values of each figure of merit.) Based on these results, we conclude that OSTD reduces the critical path delay by an average of 11% at the cost of an average of 0.7% increase in the total wire length for the benchmark circuits. The increased total wire length is caused by pushing or pulling some logic cell during the layout adjustment step needed to remove the overlaps between the sleep transistors and logic cells. The last column in Table 1 shows the runtime of OSTD algorithm for the benchmark circuits.

Table 2 shows comparison results between OSTD and USTD in terms of the critical path delay. OSTD reduces the critical path delay by an average 3.8% compared to USTD. Notice that we limited ourselves to small circuit benchmarks due to the lack of capacity by Hspice to simulate large circuits. We expect that the advantage of our proposed method (OSTD) over CSTD and USTD becomes more pronounced as the number of logic cells in the circuit increases.

Table 2 Comparisons of Circuit Performance between USTD and OSTD

Circuit	Critical Path Delay (ns)		Reduction
	USTI	OSTI	
C432	2.01	1.96	2.25%
C499	1.53	1.49	2.50%
C880	1.71	1.68	2.17%
C1355	1.95	1.86	4.53%
C1908	2.25	2.18	3.20%
C3540	4.33	4.11	5.08%
C5315	4.26	4.03	5.40%
C6288	7.47	7.15	4.31%
C7552	4.11	3.90	5.09%
Avg.			**3.84%**

6. CONCLUSION

An optimal sleep transistor cells placement methodology for MTCMOS circuits was presented. The presented algorithm provides optimal locations of sleep transistor cells for the standard cell-based layout design so that the performance degradation of MTCMOS circuit due to the interconnect resistance of the virtual ground network is minimized.

References

[1] F. Fallah and M. Pedram, "Standby and active leakage current control and minimization in CMOS VLSI circuits." *IEICE Trans. on Electronics*, Vol. E88–C, No. 4, pp. 509-519, Apr. 2005.

[2] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-v power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, pp. 847-854, Aug. 1995.

[3] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukuda, 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] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor sizing issues and tool for multi-threshold CMOS technology," in *Proc. IEEE/ACM Design Automation Conf.*, 1997, pp. 409-414.

[5] V. Khandelwal and A. Srivastava, "Leakage control through fine-grained placement and sizing of sleep transistors," in *Proc. ACM/IEEE Int. Conf. on Computer Aided Design*, 2004, pp. 533-536.

[6] B. H. Calhoun, F A. Honore, and A. Chandrakasan, "Design methodology for fine-grained leakage control in MTCMOS," in *Int. Symp. Low Power Electronics and Design*, 2003, pp.104-109.

[7] H. Won, K. Kim, K. Jeong, K. Park, K. Choi, and J. Kong, "An MTCMOS design methodology and its application to mobile computing," in *Int. Symp. Low Power Electronics and Design*, 2003, pp.110-115.

[8] J. Kao, S. Narenda, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," in *Proc. ACM/IEEE Design Automation Conf*, 1988, pp. 495-500.

[9] M. Anis, S. Areibi, and M. Elmasry, "Design and optimization of Multi-threshold CMOS circuit," *IEEE Trans. Computer-Aided Design of Integrated Circuits Systems*, vol. 22, pp. 1324-1342, Oct. 2003.

[10] S. Mutoh, S. Shigematsu, Y. Gotoh, and S. Konaka, "Design method of MTCMOS power switch for low-voltage high-speed LSIs," in *Proc. Asian and South Pacific Design Automation Conf*, 1999, pp. 113-116.

[11] K. Usami, N. Kawabe, M. Koizumi, K. Seta, and T. Furusawa, "Automated selective multi-threshold design for ultra-low standby applications," in *Int. Symp. Low Power Electronics and Design*, 2002, pp.202-206.

[12] T. Kitahara, N. Kawabe, F. Minami, K. Seta, and T. Furusawa, "Area-efficient selective multi-threshold CMOS design methodology for standby leakage power reduction," in *Proc. Design Automation and Test in Europe*, 2005, pp.646-647.

[13] C. Long and L. He, "Distributed sleep transistor network for power reduction," *IEEE Trans. Computer-Aided Design of Integrated Circuits Systems*, vol. 12, pp. 937-946, Sep. 2004.

[14] N. Ohkubo and K, Usami, "Delay modeling and static timing analysis for MTCMOS circuits," in *Proc. Asian and South Pacific Design Automation Conf*, 2006, pp. 570-575.

[15] C. Hwang, C. Kang, and M. Pedram, "Gate sizing and replication to minimize the effects of virtual ground parasitic resistances in MTCMOS designs," in *Int. Symp. Quality Electronic Design*, 2006, pp.172-177.

[16] P. Babighian, L. Benini, A. Macii, and E. Macii, "Post-layout leakage power minimization based on distributed sleep transistor insertion," in *Int. Symp. Low Power Electronics and Design*, 2004, pp.138-143.

[17] 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.

[18] C. Hwang and M. Pedram, "Timing-driven placement based on monotone cell ordering constraints," in *Proc. Asian and South Pacific Design Automation Conf*, 2006, pp. 201-206.

Table 1 Comparisons of Circuit Performance between CSTD and OSTD

Circuit	Wire length		Increase	Critical Path Delay (ns)		Reduction	CPU (sec)
	CSTI	OSTI		CSTI	OSTI		
C432	25517.1	25635.1	0.46%	2.11	1.96	6.89%	0.01
C499	40393.1	40740.5	0.86%	1.60	1.49	6.76%	0.01
C880	39895.9	40389.2	1.24%	1.84	1.68	9.10%	0.01
C1355	41055.5	41588.2	1.30%	1.98	1.86	6.23%	0.01
C1908	38859.9	39324.6	1.20%	2.35	2.18	7.16%	0.01
C3540	96423.0	96593.5	0.18%	4.92	4.11	16.39%	0.01
C5315	203559.5	204011.8	0.22%	4.87	4.03	17.13%	0.02
C6288	83024.1	83465.3	0.53%	8.03	7.15	10.96%	0.02
C7552	210163.7	210553.9	0.19%	4.90	3.90	20.31%	0.02
Avg.			**0.69%**			**11.22%**	