Simultaneous Fine-grain Sleep Transistor Placement and Sizing for Leakage Optimization

EE Department, Tsinghua University, Beijing, P.R. China
{wangyuu99, linhai99}@mails.tsinghua.edu.cn {yanghz, luorong, wangh}@tsinghua.edu.cn

Abstract

With the growing scaling of technology, leakage power dissipation has become a critical issue of VLSI circuits and systems designs. Multi-threshold CMOS leads to about 10X leakage reduction in circuit standby mode. In this paper, we reduce leakage current through fine-grain sleep transistor (ST) insertion which makes it easier to guarantee circuit functionality at high speed and improves circuit noise margins [1]. We model the leakage current reduction problem as a mixed-integer linear programming (MLP) problem in order to simultaneously choose where to add the sleep transistors and the sleep transistors’ sizes optimally. The model is solved with both continuous (MLP-C) and discrete (MLP-D) sleep transistor size constraints. Furthermore a method to speed up MLP-D model is introduced. Because of the better circuit slack utilization, our experimental results show that the MLP-C model can achieve 79.75%, 93.56%, 94.99% leakage saving when the circuit slow down is 0%, 3%, 5% respectively. The MLP-C model also achieves on average 74.79% less area penalty compared to the conventional fixed slowdown method when the circuit slowdown is 7%. The MLP-D model can achieve similar leakage saving compared to the MLP-C model. The MLP-CtoD method can speed up the MLP-D model 30X times with almost no difference in leakage reduction.

1. Introduction

With technology stepping into the submicron region, power issues have already reached a bottleneck in the design of portable and wireless electronic systems. The total power dissipation consists of dynamic power, short circuit power and leakage power, thus can be expressed as:

\[ P_{\text{total}} = P_{\text{dynamic}} + P_{\text{leakage}} + P_{\text{shortcircuit}} \]

\[ = \sum_{i=1}^{N} \left( \frac{1}{2} \alpha_i f C_i V_{\text{dd}}^2 + I_i V_{\text{dd}} + \alpha_i f Q_{\text{short}}, V_{\text{dd}} \right) \]

Where, \( f \) is the operation frequency, \( V_{\text{dd}} \) is the supply voltage, and \( N \) is the number of gates. \( \alpha_i \), \( C_i \), \( I_i \), and \( Q_{\text{short},i} \) are the transition probability, load capacitance, leakage current, and short circuit charge of the \( i \)-th gate, respectively. The behavior of the short circuit power dissipation remains at around 10% of the total power dissipation [2]. With the development of the fabrication technology, leakage power dissipation has become comparable to switching power dissipation [3]. At the 90nm technology node, leakage power may make up 42% of total power [4].

Inevitably, techniques are necessary for reducing the increasing leakage power. These leakage control methods can be broadly categorized into two main categories: process level and circuit level techniques [5]. At the process level, leakage reduction can be achieved by controlling the dimensions (length, oxide thickness, junction depth, etc.) and doping profile in transistors. Here we talk about circuit design techniques, namely, adapt body bias [6], DVTS [7], input vector control [8], dual-Vt assignment [9] [10] and Multi-Threshold CMOS (ST insertion).

![Fine-grain vs Cluster-based ST Insertion](image)

Among these, Multi-Threshold CMOS (MTCMOS) is a valuable technique for reducing leakage power in the circuit standby mode. MTCMOS technique is essentially placing a sleep transistor between the gates and the power/ground (P/G) net in order to put them into sleep mode when the circuit is standby. The most popular MTCMOS technique is gating the power of sizable blocks using large sleep transistors which assumes that all gates have a fixed slowdown [11] [12] [13] [14] [15]. However,
in recent years the use of sleep devices in the gate level [1] [16] (Figure 1. (a)), which has some advantages over the block level design (Figure 1. (b)), is raising people’s concern.

The existing literature on MTCMOS circuits [11-15] present cluster based methods for sleep transistor insertion and sizing. [11] first gives out a mutual exclusion method to reduces the area penalty. [12] [13] present several heuristic techniques for efficient gate clustering and try to mitigate the ground problem by introducing additional power penalty. In [14] [15], a Distributed Sleep Transistor Network (DISTN) approach is proposed which connects all the sleep devices to reduce the area penalty.

Although cluster based methods reduce the area penalty, they induce large ground bounce in the P/G network which has adverse effects on circuit speed and noise immunity [16]. What is more, the sleep transistor’s size is determined by the worst case current of the clustering block. However identifying the worst case is quite difficult without comprehensive simulation [11]. Thus it is harder to guarantee circuit functionality for large blocks with only one sleep transistors [1].

The fine-grain MTCMOS design methodology is discussed in [1] [16]. In [1], a fine-grain MTCMOS design methodology and several design rules are proposed. The authors also make a comparison between local and global devices. [16] presents a selectively sleep transistor insertion methodology with better utilization of circuit slack. They first select where to put the sleep transistors by a heuristic method and then solve an LP model to get optimal sleep transistor size. Although the second step can give out an optimal sizing result, the first step may lead to a local optimal point. Furthermore, in the second step they assume the sleep transistor size is continuous which is not the real case.

This paper presents three contributions to leakage reduction through fine grain sleep transistor insertion. First, we give out our newly developed leakage current model and delay model of a single gate, which are much simpler and more exact than the models in traditional fine grain sleep transistor insertion strategy. Secondly, a formal mixed-integer linear model of the leakage current reduction problem provides the designers the relations between leakage current and circuit constraints, and makes it possible to decide where to put the sleep transistors and the sizing of the sleep transistor simultaneously and optimally. The model can be solved with discrete sleep transistor size constraint which is much practical in the real life.

The paper is organized as follows. In Section 2, we give out our leakage current and delay model for a single gate. The detail of MLP model construction is proposed in Section 3. The implementation and experimental results are presented and analyzed in Section 4. In Section 5, we conclude this paper.

2. Preliminaries

First we give out our definition of leakage current and delay model. A cell-based design flow with a given cell library is used. We assumed sleep transistors with variable size which is decided by the process technology are used in our fine-grain sleep transistor insertion design. A combinational circuit is represented by a directed acyclic graph (DAG) \( G = (V, E) \). A vertex \( v \in V \) represents a CMOS gate from the given library, while an edge \((i, j) \in E\) represents a connection from vertex \(i\) to vertex \(j\). We define \(I(v), D(v)\) as the leakage current and delay of gate \(v\) respectively.

2.1 Leakage current model

The average leakage power dissipation \(P_{\text{leakage}}(G)\) of the circuit can be expressed as the product of the average leakage current and power supply voltage.

\[
P_{\text{leakage}}(G) = V_{DD} \times I(G) \tag{1}
\]

The circuit average leakage current can be calculated as the sum of the individual gate’s average leakage current. As we all know the leakage current of a CMOS gate is decided by its structure and input pattern. We define the probability of a gate \(v\) under input pattern \(IN\) as \(PB(v, IN)\). Thus the leakage current of a gate \(v\) in the circuit can be expressed as:

\[
I_l(v) = \sum_{IN} I_l(v, IN) \times PB(v, IN) \tag{2}
\]

\(I_l(v, IN)\) is the leakage current of gate \(v\) under input pattern \(IN\).

In our fine-grain sleep transistor insertion design the leakage of a gate in the circuit is also determined by whether the sleep transistor is inserted to this gate or not. For the gates without sleep transistor, we create a leakage model for \(I_l(v, IN)\) by simulating all the gates in the standard cell library under all possible input patterns. Thus the leakage current \(I^{*\text{leakage}}(v)\) can be expressed as:

\[
I^{*\text{leakage}}(v) = \sum_{IN} I_l(v, IN) \times PB(v, IN) \tag{3}
\]

On the other hand, the subthreshold leakage current \(I^{\text{sub}}(v)\) with sleep transistors are given by [17]:

\[
I^{\text{sub}}(v) = \mu_s C_{ox} (W/L) \cdot I_l \cdot V_{T} \cdot \frac{V_{T\text{high}} - V_l}{\text{sat}} (1 - e^{-\frac{V_{T\text{high}} - V_l}{V_T}}) \tag{4}
\]

where \(\mu_s\) is the N-mobility, \(C_{ox}\) is the oxide capacitance, \(V_{T\text{high}}\) is the high threshold voltage, \(V_T\) is the...
thermal voltage, \( n \) is the sub-threshold swing parameter, \((W/L)\), represents the size of the sleep transistor inserted to gate \( v \). \( V_{th} \) is decided by \((W/L)\), thus the relationship between \( I^S(v) \) and \((W/L)\), is complicated. Here we present our simplified leakage current \( I^S(v) \) model:

\[
I^S(v) = A(v) + B(v) \times (W/L),
\]

where \( A(v) \), \( B(v) \) are constants and are decided by the gate type.

Consider two standard cells: a two-input NAND and a four-input AND with fixed structure and size in the given library. We add high threshold voltage sleep transistor to the gates, and compare the leakage current of the gates with different sleep transistor sizes. Refer to our model, we can give out the \( A(v) \), \( B(v) \) of the NAND2 and AND4 respectively: 1.31774, 0.01128; 1.67104, 0.01514.

### Table 1 Leakage current with different sleep transistor sizes in NAND2 and AND4

<table>
<thead>
<tr>
<th>( W/L )</th>
<th>Leakage current in NAND2 (pA)</th>
<th>Leakage current in AND4 (pA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.31774, 0.01128</td>
<td>1.67104, 0.01514</td>
</tr>
<tr>
<td>4</td>
<td>1.50123, 0.0162</td>
<td>1.70216, 0.0103</td>
</tr>
<tr>
<td>8</td>
<td>1.40795, 0.0104</td>
<td>1.70216, 0.0103</td>
</tr>
<tr>
<td>16</td>
<td>1.40998, 0.0104</td>
<td>1.70421, 0.0103</td>
</tr>
</tbody>
</table>

Notice \( I^S(v) \) is still sensitive to the input patterns, the data shown in Table 1 is the average leakage current for which we assume all the input patterns have same probability. As shown in Table 1, the error is less than 0.39% and the original leakage current without sleep transistor is at least 15X larger than \( I^S(v) \). We estimate every \( A(v) \), \( B(v) \) for all the standard cells and find out, on average, \( B(v) \)'s are around 1% of \( A(v) \), thus the variation range of \( I^S(v) \) is about 15% of \( A(v) \).

Thus we use lookup table to model the leakage current of gates with no sleep transistor, and linear equations to model the leakage current of gates with sleep transistors. As we can see, our leakage current model for a single gate is very simple and accurate.

### 2.2 Delay model

In our fine-grain sleep transistor insertion design, we have to insert sleep transistors to the original gates in the given library. As shown in [18], the delay of the gate is influenced by the sleep transistor insertion. The load dependent delay \( D^{v,v}(v) \) of gate \( v \) without sleep transistors can be expressed as:

\[
D^{v,v}(v) = \frac{KC_v V_{th}}{(V_{th} - V_{thmin})^2}
\]

where \( C_v \), \( V_{thmin} \), \( K \) are the load capacitance at the gate output, the low threshold voltage, the velocity saturation index and the proportionality constant respectively. The propagation delay \( D^S(v) \) with the presence of sleep transistors of gate \( v \) can be expressed as:

\[
D^S(v) = \frac{KC_v V_{th}}{(V_{th} - 2V_s - V_{thmin})^2}
\]

where \( V_s \) is the \( V_{th} \) of the sleep transistor, that is to say the voltage drop from \( V_{th} \) to the virtual \( V_{th} \) as shown in Figure 1. We define the difference of \( D^{v,v}(v) \) and \( D^S(v) \) as \( \Delta D(v) \):

\[
\Delta D(v) = D^S(v) - D^{v,v}(v)
\]

Refer to equation (6) (7) (8), we can get an approximate \( \Delta D(v) \) with neglectable difference using Taylor series expansion:

\[
\Delta D(v) = D^S(v) - D^{v,v}(v) = \left((1 + \frac{2V_s}{V_{th} - V_{thmin}} + \frac{2V_s^2}{2(V_{th} - V_{thmin})^2} + \cdots) - 1\right)\text{\( D^{v,v}(v) \)}
\]

Thus we use \( I^2=2\alpha(V_{DD} - V_{Thlow}) \) to simplify the equation (9) since \( V_{Thlow} \), \( \alpha \) are all technology dependent constant. We suppose \( I_{on}(v) \) is the current flowing through the sleep transistor in the gate \( v \) during the active mode, and can be expressed as [16]:

\[
I_{on}(v) = \mu C_{xx}(W/L) \times \left| V_{th} - V_{thmin} \right| W \cdot \frac{V_s^2}{2}
\]

Thus the voltage drop \( V_s \) in gate \( v \) due to sleep transistor insertion can be expressed as:

\[
V_s = \frac{I_{on}(v)}{\mu C_{xx}(W/L) \times \left| V_{th} - V_{thmin} \right| W}
\]

Here we use \( \Psi(W) \) to simplify the equation. From above we can get \( \Delta D(v) \) as:

\[
\Delta D(v) = \left(\Psi(W) \times (W/L) \right)^{-1} \times \left(1 + \frac{\alpha + 1}{\alpha \Psi(W) \times (W/L)} \right)^{-1}
\]

From equation (10), we can see \( V_s \) is slightly larger than the actual value, thus \( \Delta D(v) \) is a little bit larger than the actual value which make our model more feasible to maintain the timing constraints of the circuit.

### 3. MLP model construction

We now construct an MLP model for simultaneous placement and sizing of sleep transistor. There are only two conditions to each gate \( v \) with or without sleep transistor. We therefore define a binary variable \( ST(v) \) to represent gate \( v \)'s sleep transistor condition, where \( ST(v) = 1 \) for gate \( v \) with sleep transistor inserted and \( ST(v) = 0 \) for gate \( v \) without sleep transistor.

#### 3.1 Objective function
We use equation (3) as basis to construct the objective function. Note that the leakage current of gate \( v \) \( I_L(v) \) can be written as:

\[
I_L(v) = I^0_L(v) + (1 - ST(v)) + I^{ST}_L \times ST(v) \tag{13}
\]

Therefore we represent the total leakage current by:

\[
I(G) = \sum_{v} I^0_L(v) + (1 - ST(v)) + I^{ST}_L \times ST(v) \tag{14}
\]

Refer to equation (3) and (5), we can hence replace equation (13) with:

\[
I(G) = \sum_{v} \left( \sum_{IN} I_L(v, IN) \times PB(v, IN) \times (1 - ST(v)) \right) \tag{15}
\]

where \( ST(v) \) and \( (W/L) \), are variables which decide where to put sleep transistor and how to size the sleep transistor respectively.

3.2 Timing constraints

First we consider the primary input (PIs) and output (POs) gates of the circuit. The arrival time \( t_a \) of all the PIs are set to zero, while the required time of all the POs are less than the overall circuit delay \( T_{req} \):

\[
\begin{align*}
t_a(m) &= 0 & m & \in PI \tag{16} \\
t_a(n) + D(n) &\leq T_{req} & n & \in PO \tag{17}
\end{align*}
\]

Then we notice that the sum of gate \( v \)'s arrival time and its delay must be smaller than the arrival time of gate \( v \)'s fanout gates. That is to say, \( \forall (i, j) \in E, i, j \in V \), we can derive the constraint as:

\[
t_a(i) + D(i) \leq t_a(j) \tag{18}
\]

As we have already induced the definition of \( ST(v) \), we can rewrite the delay of gate \( v \) as:

\[
D(v) = D^+(v) + \Gamma \Psi D^-(v) \times (W/L) \times ST(v) \tag{19}
\]

3.3 Linearization constraints

First we define variable \( W(v) \) for each gate, where \( WL(v) = (W/L) = 2^{m(v)} \), \( WLN(v) = (W/L)^2 = 2^{m(v)} \), \( WLN2(v) = (W/L)^2 = 2^{m(v)} \), and \( V \in [0, W_{max}] \). We use a similar piecewise linear approximation technique in [19] to linearize these exponential expressions with inequalities:

\[
\begin{align*}
WL(v) &\geq 2^k W(v) + (1-k) \times 2^i, & k = 0, 1, \ldots, W_{max} \\
WLN(v) &\geq -2^k W(v) + (1-k) \times 2^i, & k = -W_{max}, -W_{max} + 1, \ldots, 0 \\
WLN2(v) &\geq -2^k W(v) + (1-k) \times 2^i, & k = -2W_{max}, -2(W_{max} + 1), \ldots, 0
\end{align*}
\]

Secondly, in equation (15) and (19), a set of items to be linearized is:

\[
WS(v) = (W/L) \times ST(v) = WL(v) \times ST(v) \tag{13}
\]

\[
WSN(v) = (W/L)^2 \times ST(v) = WLN(v) \times ST(v) \tag{14}
\]

\[
WSN2(v) = (W/L)^2 \times ST(v) = WLN2(v) \times ST(v) \tag{15}
\]

where \( WL(v), WLN(v), WLN2(v) \) are real variables while \( ST(v) \) is binary. As in [19], \( C = B \times A \), where \( A \) is a binary variable and \( M \) is an upper bound of \( B \), is linearized as follows:

\[
\begin{align*}
0 &\leq C \leq B \\
C &\leq M \times A \\
C &\geq B - M(1-A)
\end{align*}
\]

Since \( W(v) \in [0, W_{max}] \), \( WL(v), WLN(v) \) and \( WLN2(v) \) all have their upper bound. Hereeto, we end up our MLP model for leakage minimization. And the general form of our MLP model is given out in Figure 2.

3.4 MLP model with discrete size constraint

In our MLP model presented in Figure 2, \( W(v) \) is a continuous real variable which is not the real case. Thus we add a constraint that the \( W(v) \)'s are integers, which means the sizes of the sleep transistors are powers of two. It is clear we can change the constraints to fit other discrete conditions of sleep transistors’ sizes. We name the MLP model with continuous size constraints as MLP-C, the MLP model with integer size constraints as MLP-D.

4. Implementation and experiment results

We use ISCAS85 benchmark circuits to evaluate our MLP model. The netlists are synthesized using Synopsys Design Compiler and a TSMC 0.18\( \mu \text{m} \) standard cell library. The leakage current look up table is generated by HSPICE with TSMC 0.18\( \mu \text{m} \) CMOS process and a 1.8v supply condition. The values of various transistor parame-
named solved by various LP solvers, here we use an LP solver furthermore our MLP models. The MLP models can be automatically generate the timing information and specialized static timing analysis (STA) tool [10] to result with all the is larger than 6%, our MLP-C model can get a optimal and fixed slowdown method in the same circuit slowdown difference of leakage saving between our MLP-C model that mentioned in [16]. In our experimental results, the conventional fixed slowdown method is not as large as difference of the leakage saving between our model and 7% or 9% circuit slowdown condition. However, the slowdown condition than fixed slowdown method in the model can achieve more leakage saving in the 5% circuit slowdown condition than fixed slowdown method. On the other hand our MLP model can not get a valid solution through conventional fixed slowdown method. As we can see, our MLP-C model can achieve more leakage saving in the 5% circuit slowdown condition than fixed slowdown method in the 7% or 9% circuit slowdown condition. However, the difference of the leakage saving between our model and conventional fixed slowdown method is not as large as that mentioned in [16]. In our experimental results, the difference of leakage saving between our MLP-C model and fixed slowdown method in the same circuit slowdown condition is within 11%. That is caused by the difference leakage current model. When the performance slowdown is larger than 6%, our MLP-C model can get a optimal result with all the ST(v)=1, which leads to the same result as optimal sizing with sleep transistors placed everywhere [16].

Table 2 Leakage current saving through MLP-C Model and Fixed-Slowdown Method

<table>
<thead>
<tr>
<th>ISCAS85 benchmark circuits</th>
<th>Original Ileak (pA)</th>
<th>0% Fixed-Slowdown Ileak (pA)</th>
<th>3% Fixed-Slowdown Ileak (pA)</th>
<th>5% Fixed-Slowdown Ileak (pA)</th>
<th>0% MLP-C (pA)</th>
<th>3% MLP-C (pA)</th>
<th>5% MLP-C (pA)</th>
<th>0% Fixed Slowdown (pA)</th>
<th>3% Fixed Slowdown (pA)</th>
<th>5% Fixed Slowdown (pA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C412</td>
<td>3519.96</td>
<td>257.01</td>
<td>322.96</td>
<td>328.39</td>
<td>352.39</td>
<td>352.39</td>
<td>352.39</td>
<td>352.39</td>
<td>352.39</td>
<td>352.39</td>
</tr>
<tr>
<td>C499</td>
<td>1236.63</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
<td>1231.76</td>
</tr>
<tr>
<td>C313</td>
<td>2592.97</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
<td>2242.31</td>
</tr>
<tr>
<td>C1908</td>
<td>3812.12</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
<td>3293.78</td>
</tr>
<tr>
<td>C3540</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
<td>3298.34</td>
</tr>
<tr>
<td>C3515</td>
<td>1519.31</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
<td>1427.42</td>
</tr>
<tr>
<td>C2688</td>
<td>5775.65</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
<td>10746</td>
</tr>
</tbody>
</table>

Leakage saving

Table 3 Comparison between MLP-C and Fixed-slowdown

<table>
<thead>
<tr>
<th>ISCAS85 benchmark circuits</th>
<th>ST area (W/L)</th>
<th>Ileak (pA)</th>
<th>ST area (W/L)</th>
<th>Ileak (pA)</th>
<th>ST area (W/L)</th>
<th>Ileak (pA)</th>
<th>ST area (W/L)</th>
<th>Ileak (pA)</th>
<th>ST area (W/L)</th>
<th>Ileak (pA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C412</td>
<td>321.83</td>
<td>174.45</td>
<td>259.04</td>
<td>231.17</td>
<td>259.04</td>
<td>231.17</td>
<td>259.04</td>
<td>231.17</td>
<td>259.04</td>
<td>231.17</td>
</tr>
<tr>
<td>C499</td>
<td>123.02</td>
<td>71.84</td>
<td>109.44</td>
<td>93.35</td>
<td>109.44</td>
<td>93.35</td>
<td>109.44</td>
<td>93.35</td>
<td>109.44</td>
<td>93.35</td>
</tr>
<tr>
<td>C313</td>
<td>338.81</td>
<td>1014.09</td>
<td>933.54</td>
<td>869.91</td>
<td>933.54</td>
<td>869.91</td>
<td>933.54</td>
<td>869.91</td>
<td>933.54</td>
<td>869.91</td>
</tr>
<tr>
<td>C1908</td>
<td>1334.86</td>
<td>2534.59</td>
<td>1527.64</td>
<td>1493.71</td>
<td>1527.64</td>
<td>1493.71</td>
<td>1527.64</td>
<td>1493.71</td>
<td>1527.64</td>
<td>1493.71</td>
</tr>
<tr>
<td>C3540</td>
<td>2558.78</td>
<td>2814.23</td>
<td>2558.78</td>
<td>2558.78</td>
<td>2558.78</td>
<td>2558.78</td>
<td>2558.78</td>
<td>2558.78</td>
<td>2558.78</td>
<td>2558.78</td>
</tr>
<tr>
<td>C3515</td>
<td>2693.62</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
<td>3575.65</td>
</tr>
<tr>
<td>C2688</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
<td>3623.68</td>
</tr>
<tr>
<td>C4288</td>
<td>3068.19</td>
<td>11732.32</td>
<td>4042.77</td>
<td>3353.86</td>
<td>4042.77</td>
<td>3353.86</td>
<td>4042.77</td>
<td>3353.86</td>
<td>4042.77</td>
<td>3353.86</td>
</tr>
</tbody>
</table>

Leakage saving

Table 4 Comparison of MLP-C, MLP-CtoD and MLP-D

In Table 3, we compare the area penalty between MLP-C model and fixed slowdown method. As we mentioned above, the difference of leakage saving is not very large. However, our MLP-C model can achieve a much less sleep transistor area penalty. With 7% circuit slowdown, our MLP-C model leads to 74.79% sleep transistor area saving compared to fixed slowdown method.

Obviously, an MLP-D model is very time-consuming because of the integer constraints and the increasing circuit size. Thus we derived a fast method (MLP-CtoD) to solve it based on the MLP-C model. The circuit produced by this continuous sleep transistor size scenario provides us a set of W(v)<sup>C</sup> and ST(v)<sup>C</sup>. We directly choose the sleep transistor size W(v)<sub>D</sub> = Ceiling(W(v)<sub>C</sub>)<sup>VTHhigh</sup>, and ST(v)<sub>D</sub> = Floor(ST(v)<sub>C</sub>). We directly put the output of the feedback gate structure [21] in order to avoid floating states. Meanwhile the results for the area penalty imposed by the fine-grain sleep transistor in [16] show that the area
penalty is just around 5% through a standard cell placement methodology.

5. Conclusions

We have presented a mixed integer linear programming method to simultaneously place and size the sleep transistor in our fine-grain sleep transistor design to minimize the leakage current. A novel leakage current and delay model of the fine-grain sleep transistor design is presented in order to build up the MLP model. Our MLP model can minimize the leakage current to about 79.75% even though the circuit performance is not influenced. Two MLP model: MLP-C and MLP-D with different sleep transistor size constraints are presented and compared. The MLP-D uses a discrete sleep transistor size constraint which is more practical. An MLP-CtoD method is introduced to speed up MLP-D model and approximate the MLP-D model very well. Our method is introduced to speed up MLP-D model and minimize the leakage power. A novel leakage current modeling and reduction techniques, in

Table 4 Comparison of MLP-C, MLP-CtoD and MLP-D

<table>
<thead>
<tr>
<th>Circuit slowdown</th>
<th>MLP-C</th>
<th>MLP-CtoD</th>
<th>MLP-D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L mop (pA)</td>
<td>ST Area (W/L)</td>
<td>ST Gate</td>
</tr>
<tr>
<td>0%</td>
<td>2177.01</td>
<td>489.36</td>
<td>137.169</td>
</tr>
<tr>
<td>5%</td>
<td>541.28</td>
<td>532.81</td>
<td>147.169</td>
</tr>
<tr>
<td>10%</td>
<td>785.5</td>
<td>585.31</td>
<td>157.169</td>
</tr>
<tr>
<td>15%</td>
<td>1297.05</td>
<td>532.42</td>
<td>167.169</td>
</tr>
<tr>
<td>20%</td>
<td>397.01</td>
<td>596.45</td>
<td>177.169</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit slowdown</th>
<th>MLP-C</th>
<th>MLP-CtoD</th>
<th>MLP-D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L mop (pA)</td>
<td>ST Area (W/L)</td>
<td>ST Gate</td>
</tr>
<tr>
<td>0%</td>
<td>2177.01</td>
<td>489.36</td>
<td>137.169</td>
</tr>
<tr>
<td>5%</td>
<td>541.28</td>
<td>532.81</td>
<td>147.169</td>
</tr>
<tr>
<td>10%</td>
<td>785.5</td>
<td>585.31</td>
<td>157.169</td>
</tr>
<tr>
<td>15%</td>
<td>1297.05</td>
<td>532.42</td>
<td>167.169</td>
</tr>
<tr>
<td>20%</td>
<td>397.01</td>
<td>596.45</td>
<td>177.169</td>
</tr>
</tbody>
</table>


6. Conclusions

We have presented a mixed integer linear programming method to simultaneously place and size the sleep transistor in our fine-grain sleep transistor design to minimize the leakage current. A novel leakage current and delay model of the fine-grain sleep transistor design is presented in order to build up the MLP model. Our MLP model can minimize the leakage current to about 79.75% even though the circuit performance is not influenced. Two MLP model: MLP-C and MLP-D with different sleep transistor size constraints are presented and compared. The MLP-D uses a discrete sleep transistor size constraint which is more practical. An MLP-CtoD method is introduced to speed up MLP-D model and approximate the MLP-D model very well. Our experimental results show that the MLP-C model can achieve 93.56%, 94.99% leakage saving when the circuit slowdown is 3%, 5% respectively. The MLP-C model also achieve on average 74.79% less area penalty compared to the conventional fixed slowdown method when the circuit slowdown is 7%. The MLP-D model can also achieve just 0.1% less leakage saving compared to the MLP-C model. The MLP-CtoD method can speed up the MLP-D model 30X times within almost no difference in leakage reduction.

6. Reference


[20] [http://groups.yahoo.com/group/lp_solve/](http://groups.yahoo.com/group/lp_solve/)