Coarse-Grain MTCMOS Sleep Transistor Sizing Using Delay Budgeting

Ehsan Pakbaznia and Massoud Pedram

University of Southern California
Dept. of Electrical Engineering
DATE-08 Munich, Germany
Leakage in CMOS Technology

- $V_{dd}$ is reduced with CMOS technology scaling.
- $V_{th}$ must be lowered to recover the transistor switching speed.
- The subthreshold leakage current increases exponentially with decreasing $V_{th}$.
- A highly effective leakage control mechanism has proven to be the MTCMOS technique.
Overview of MTCMOS

- A high-$V_{th}$ transistor is used to disconnect low-$V_{th}$ transistors from the ground or the supply rails.
Coarse-Grain MTCMOS

- Coarse-grain vs. fine-grain:
  - Smaller sleep transistor area
  - Lower leakage
  - Regular standard cell library can be used (no need to characterize new cells)
Sleep Transistor Layout

Single transistor footer switch

Single transistor header switch

Double-transistor (mother/daughter) footer switch
Sleep Transistor Placement

- Symmetric placement styles are preferred due to lower routing complexity for TVDD/TVSS and SLEEP/SLEEPB signals
Notion of Module

- \((r,i)\) denotes the module that is formed around the \(i^{th}\) sleep transistor in the \(r^{th}\) row of the standard cell layout.
- The cells belonging to \((r,i)\) are those that are in the \(r^{th}\) row and are closest in distance to the \(i^{th}\) sleep transistor in that row.
Time-dependant Current Source Model for Modules

- VVSS rail resistance between the cells inside each module is ignored.
- $r_{\text{VSS}(r,i)}$ denotes the VVSS resistance between modules $(r,i)$ and $(r,i+1)$.
- $I_{M(r,i)}(t)$ and $I_{\text{st}(r,i)}(t)$ denote the module discharging current and the sleep transistor current of module $(r,i)$. 
Motivational Example

- Circuit: FO4 inverter chain
- Modules: M1 and M2
- Sleep Transistors: replaced by their linear resistive models, $R_1$ and $R_2$
- CMOS ($R_1=R_2=0$) delay: 103ps

<table>
<thead>
<tr>
<th>Module</th>
<th>Module Delay (pico sec)</th>
<th>Module Peak Current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>46</td>
<td>0.3</td>
</tr>
<tr>
<td>$M_2$</td>
<td>57</td>
<td>4.65</td>
</tr>
</tbody>
</table>
Effect of Slack Distribution on Total Sleep Transistor Size

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Module Delay (ps)</th>
<th>Total Delay (ps)</th>
<th>Sleep Tx Resistance (Ω)</th>
<th>$\sum R_i^{-1}$ (Ω⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>$T_{M1} = 46$</td>
<td>103</td>
<td>$R_1 = 0$</td>
<td>$R_2 = 0$</td>
</tr>
<tr>
<td></td>
<td>$T_{M2} = 57$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTCMOS</td>
<td>$T_{M1} = 50.6$</td>
<td>113.3</td>
<td>$R_1 = 250$</td>
<td>$R_2 = 9$</td>
</tr>
<tr>
<td></td>
<td>$T_{M2} = 62.7$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{M1} = 52$</td>
<td>113.3</td>
<td>$R_1 = 330$</td>
<td>$R_2 = 2$</td>
</tr>
<tr>
<td></td>
<td>$T_{M2} = 61.3$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{M1} = 48$</td>
<td>113.3</td>
<td>$R_1 = 110$</td>
<td>$R_2 = 25$</td>
</tr>
<tr>
<td></td>
<td>$T_{M2} = 65.3$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Total available slack: 10.3ps (10% delay penalty)
  - **Case 1**: uniformly distributed slack *(medium)*
  - **Case 2**: 80% for M1 and 20% for M2 *(worst)*
  - **Case 3**: 20% for M1 and 80% for M2 *(best)*

- Current-aware optimization: must slow down modules with larger discharge current more
Delay-Budgeting Constraints for Sizing

- Delay-budgeting constraints: non-negative slack for all nodes

\[ s'_n = \left[ \min \left\{ r_{\text{fanouts of } C_n} \right\} - d'_n \right] - \left[ \max \left\{ a_{\text{fanins of } C_n} \right\} + d'_n \right] \geq 0 \]

- \( d'_n \) is the delay for cell \( C_n \in M_i \) with VVSS voltage \( v_i \). We can show:

\[ d'_n = d_n + \frac{v_i}{V_{DD} - V_{IL}} d_n \]

- To simplify the constraints we only consider the timing critical paths ➔ need to define the notion of path delay!
The delay increase for path $\pi_k$ is the summation of delay increases for all the gates in $\pi_k$:

$$\Delta d_{\pi_k} = \sum_{C_n \in \pi_k} \Delta d_n = \sum_{C_n \in \pi_k} \frac{R_{st\theta(C_n)} I_{st\theta(C_n)}^{\max} [t_{C_n}^{c_{\min}}, t_{C_n}^{c_{\max}}]}{V_{DD} - V_{tL}} d_n$$

- $\theta(C_n)$ is the index of the module that cell $C_n$ belongs to.
- $R_{st_i}$ is the linear resistance value for $i^{th}$ sleep transistor.
- $R_{st_i}$ is inversely proportional to $W_{st_i}$ (width).
- $I_{st_i}^{\max} [t_{C_n}^{c_{\min}}, t_{C_n}^{c_{\max}}]$ is the max current value flowing through $R_{st_i}$ during the time window $[t_{C_n}^{c_{\min}}, t_{C_n}^{c_{\max}}]$ when cell $C_n$ is switching.
Module Current Example

- The module current is the time-indexed summation of the expected currents for all the cells inside the module.

Current profile for a module with 3 cells and time windows:
- C1: [40, 60]
- C2: [60, 80]
- C3: [50, 70]
Delay-Budgeting (DB) Sizing Problem

Clock cycle is divided into $N$ equal time intervals. $t_j$ is the beginning time of the $j^{th}$ interval. $I_{M_i}(t_j)$ is the switching current of module $M_i$ at time $t_j$.

Minimize $\sum_{i=1}^{M} R_{st_i}^{-1}$

s.t.:
1. $\Delta d_{\pi_k} = \sum_{C_{i_k} \in \pi_k} \frac{R_{st_i} I_{st_j}^{\max \left[ C_{i_{\min}}, C_{i_{\max}} \right]}}{V_{DD} - V_{th}} d_n \leq DDR_{\text{MAX}} \times d_{\text{max}}$
   \hspace{1cm} 1 \leq k \leq K,
   \hspace{1cm} 1 \leq i \leq N

2. $R_{st_i} I_{st_i}(t_j) \leq \text{VSS}_{\text{MAX}}$; \hspace{0.5cm} 1 \leq i \leq M, \hspace{0.5cm} 1 \leq j \leq N

where:

$\forall i, j : I_{st_0}(t_j) = I_{st_{N+1}}(t_j) = 0$ and

$I_{st_i}(t_j) = I_{M_i}(t_j) + \frac{R_{st_{i-1}} I_{st_{i-1}}(t_j)}{r_{VSS_{i-1}}} + \frac{R_{st_{i+1}} I_{st_{i+1}}(t_j)}{r_{VSS_i}} - \frac{R_{st_i} I_{st_i}(t_j)}{r_{VSS_{i-1}}} - \frac{R_{st_i} I_{st_i}(t_j)}{r_{VSS_i}}$
BCM and MCM

The delay-budgeting constraints can be written as:

$$\sum_{i=1}^{M} a_{ki} R_{st_i} \leq \text{DDR\_MAX} \times d_{\text{max}}; \quad 1 \leq k \leq K$$

**Definition 1**- At any given step of the sizing algorithm, the **most critical module (MCM)** is the module with the maximum delay contribution in the $K$ most critical paths:

$$MCM = \arg \max_{M_i} \sum_{k=1}^{K} a_{ki} R_{st_i}$$

**Definition 2**- At any given step of the sizing algorithm the **best candidate module (BCM)** is defined as the module whose sleep transistor upsizing by a certain percentage will result in the largest delay improvement for unsatisfied paths.

One can show:

$$BCM = MCM \left( \left\{ \pi_k \mid 1 \leq k \leq K, \Delta d_{\pi_k} / d_{\pi_k} > \text{DDR\_MAX} \right\} \right)$$
Current-Aware Optimization

Definition 3- Least-cost BCM (LBCM) is the BCM whose sleep transistor upsizing will result in the minimum increase in the objective function.

Lemma- LBCM can be calculated as:

\[ LBCM = \arg \min_{M_i = BCM} \sum_{k=1}^{K} a_{ki} \]

\[ \Delta d_{\pi_k} > DDR_{MAX} \times d_{max} \]

At each step of the algorithm, this lemma makes the proposed algorithm a current-aware optimization algorithm.
Algorithm (step 1)

- Step 1- Initialization (NM constraints)

**Algorithm:** Slp_Initialize(I_{Mi}(t), VVSS\_MAX)

1: /*Initializing variables*/
2: for \( i=1 \) to \( M \) do
3: \( R_{st_i} = R_{MAX} \);
4: end for
5: calculate \( I_{st_i}(t_j) \) and \( v_i(t_j) = R_{st_i}I_{st_i}(t_j) \) for all \( i, j \);
6: while \( (v_i(t_j) > VVSS\_MAX \text{ for some } i \text{ or } j) \) do
7: \( M_m = \text{FindMinModule\{VVSS\_MAX - v_i(t_j)\}} \);
8: \( R_{st_m} = VVSS\_MAX/I_{st_m}(t_j) \) for all \( j \);
9: update \( I_{st_i}(t_j) \) and \( v_i(t_j) = R_{st_i}I_{st_i}(t_j) \) for all \( i, j \);
10: end while
11: return \( R_{st_i} \) for all \( i \);
Algorithm (step 2)

- Step 2- Optimization (DB constraints)

**Algorithm:** Slp_Sizing($R_{sti}$-initial, $I_M(t)$, VVSS_MAX)

1: calculate $I_{st_i}(t_j)$ and $v_i(t_j) = R_{st_i}$-initial$I_{st_i}(t_j)$ for all $i, j$;

2:  while (min_slack < 0)
3:      find LBCM and $m$=LBCM;
4:      $R_{st_m} = R_{st_m} - \alpha R_{st_m}$ ;
5:      update $I_{st_i}(t_j)$ and $v_i(t_j) = R_{st_i}I_{st_i}(t_j)$ for all $i, j$;
6:      min_slack = \infty;
7:      for $k=1$ to $K$, $j=1$ to $N$
8:         if $(\Delta d_{\pi_k} - DDR\_MAX < \text{min}\_slack)$
9:            min_slack = $\Delta d_{\pi_k} - DDR\_MAX \times d_{\text{max}}$;
10:         end if
11:     end for
12: end while
13: return ($R_{st_i}$) for all $i$;
Simulation Approach

- Max delay degradation ratio, DDR_MAX=10%
- Virtual rail resistance, $r_{VSS_i} = 0.1\Omega$
- Max number of the critical paths, K=100
- Resistance decrement factor, $\alpha = 0.1$

<table>
<thead>
<tr>
<th>Circuit</th>
<th># of cells</th>
<th># of Footers</th>
<th>Total sleep TX width ($\lambda$)</th>
<th>Proposed vs. [X] (%)</th>
<th>Proposed vs. [Y] (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>[X]=[Chiou-DAC'06] [Y]=[Chiou-DAC'07] Proposed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C17</td>
<td>7</td>
<td>2</td>
<td>53 44 16</td>
<td>70 64</td>
<td></td>
</tr>
<tr>
<td>9sym</td>
<td>276</td>
<td>30</td>
<td>786 715 312</td>
<td>60 56</td>
<td></td>
</tr>
<tr>
<td>C432</td>
<td>214</td>
<td>30</td>
<td>811 665 343</td>
<td>57 48</td>
<td></td>
</tr>
<tr>
<td>C880</td>
<td>467</td>
<td>55</td>
<td>1290 1173 579</td>
<td>55 51</td>
<td></td>
</tr>
<tr>
<td>C1355</td>
<td>546</td>
<td>60</td>
<td>1437 1597 727</td>
<td>49 54</td>
<td></td>
</tr>
<tr>
<td>C3540</td>
<td>1307</td>
<td>280</td>
<td>3920 3469 1679</td>
<td>57 52</td>
<td></td>
</tr>
<tr>
<td>C5315</td>
<td>1783</td>
<td>320</td>
<td>5799 5631 3372</td>
<td>42 40</td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td></td>
<td>2.0 1.89 1</td>
<td>55 52</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- A new sleep transistor sizing approach is proposed
- The algorithm takes a max circuit slowdown factor and produces the sizes of various sleep transistors while considering the DC parasitics of the virtual ground
- The problem can be formulated as a sizing with delay-budgeting and solved efficiently using a heuristic sizing algorithm
- The algorithm approaches the optimum solution by slowing down the modules with larger amount of discharging current more than the ones with smaller amount of discharging current, *current-aware optimization*
- The proposed technique uses at least 40% less total sleep transistor width compared to other approaches