Coarse-Grain MTCMOS Sleep Transistor Sizing Using Delay Budgeting

Ehsan Pakbaznia and Massoud Pedram

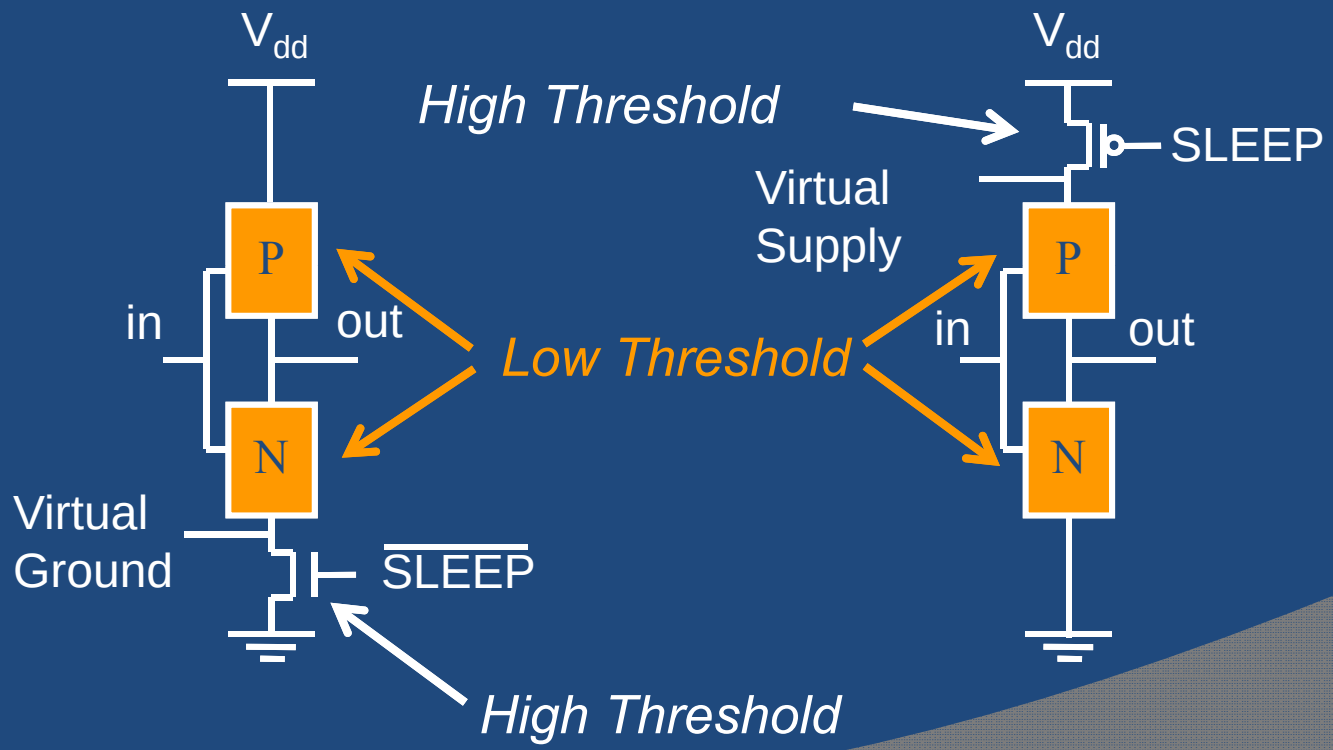
University of Southern California
Dept. of Electrical Engineering
DATE-08 Munich, Germany

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

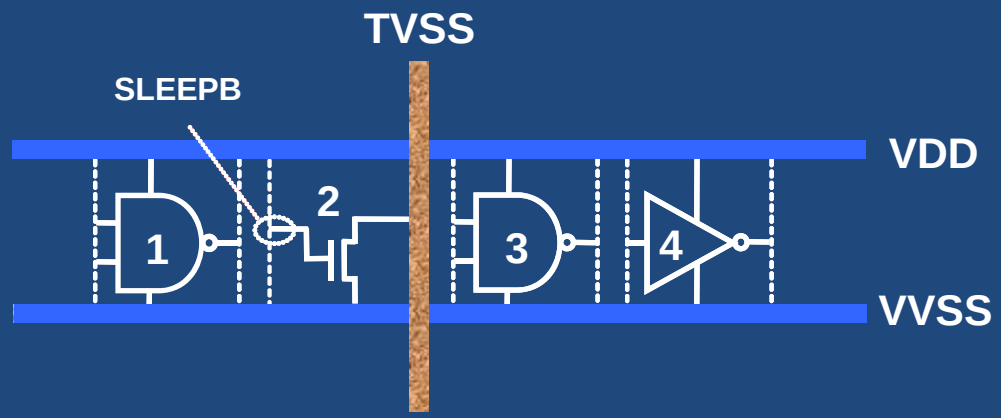
Overview of MTCMOS

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

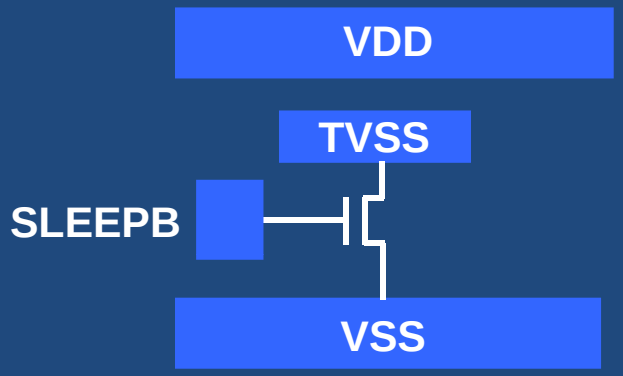


Coarse-Grain MTCMOS

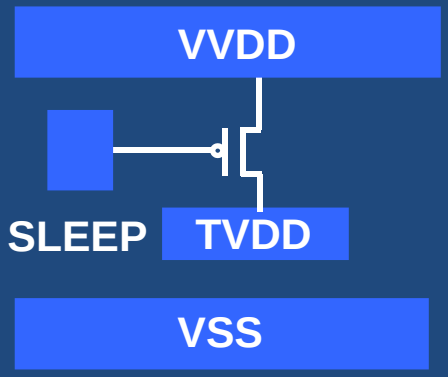
- Coarse-grain vs. fine-grain:
  - Smaller sleep transistor area
  - Lower leakage
  - Regular standard cell library can be used (no need to characterize new cells)



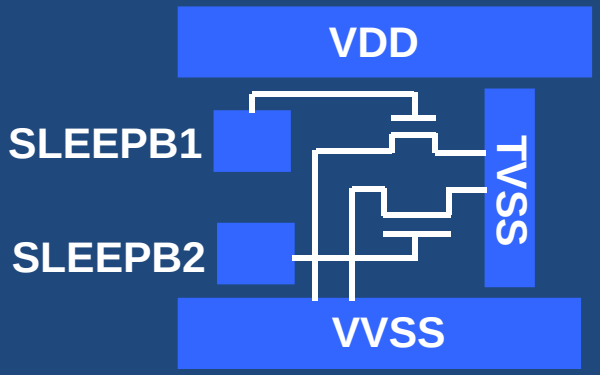
Sleep Transistor Layout



Single transistor footer switch



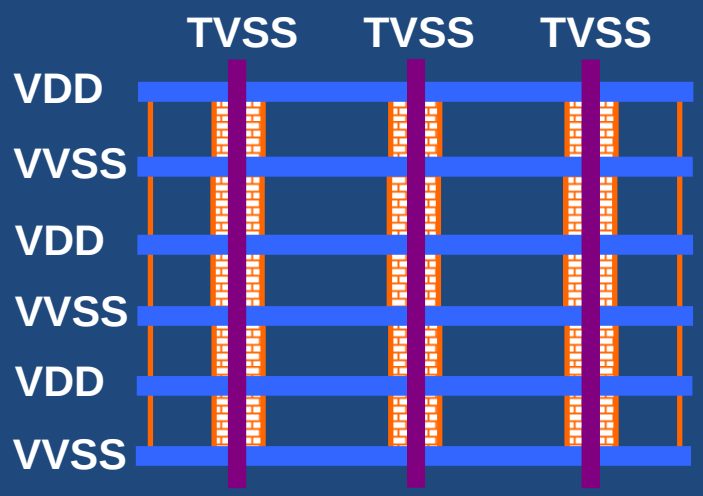
Single transistor header switch



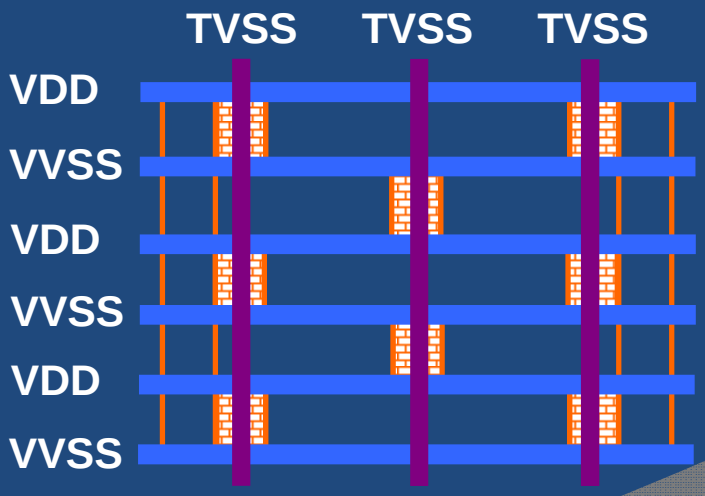
Double-transistor
(mother/daughter) footer switch

Sleep Transistor Placement

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



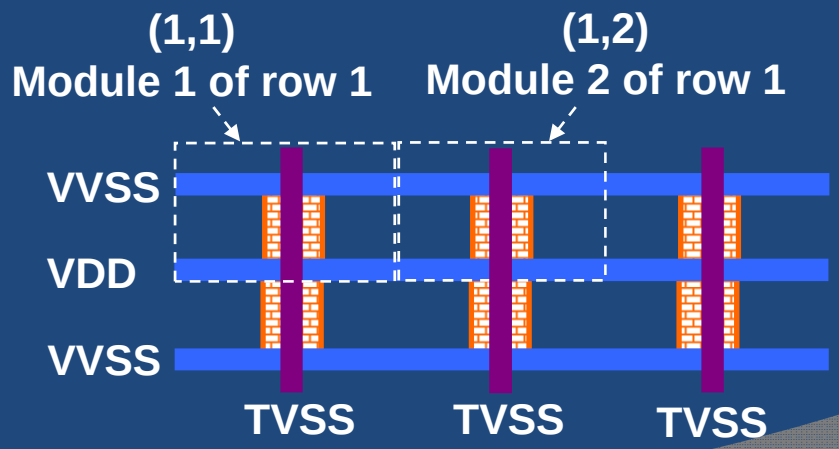
Column-based



Staggered

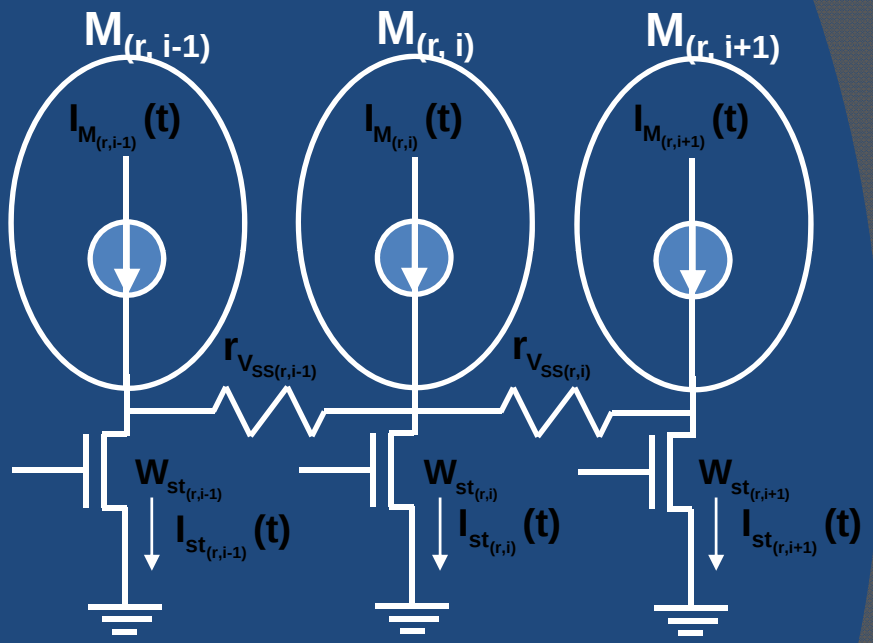
Notion of Module

- *(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



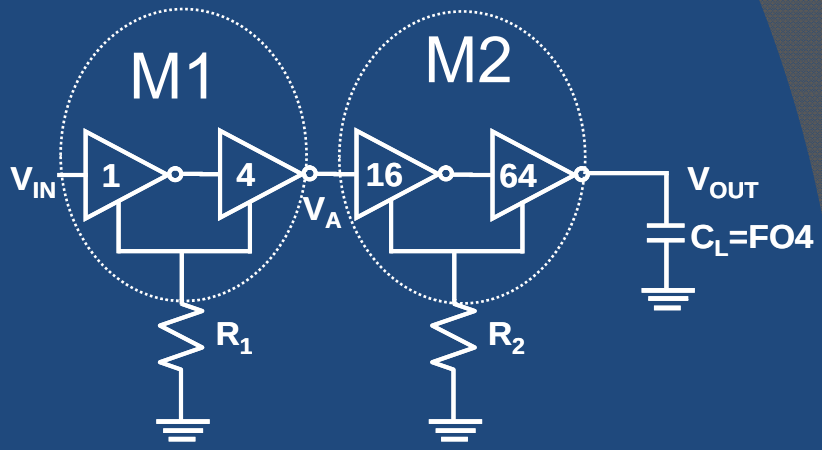
Time-dependant Current Source Model for Modules

- VVSS rail resistance between the cells inside each module is ignored
- $r_{VSS_{(r,i)}}$ denotes the VVSS resistance between modules (r,i) and (r,i+1)
- $I_{M_{(r,i)}}(t)$ and $I_{st_{(r,i)}}(t)$ 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



Module	Module Delay (pico sec)	Module Peak Current (mA)
M_1	46	0.3
M_2	57	4.65

Effect of Slack Distribution on Total Sleep Transistor Size

Circuit	Module Delay (ps)	Total Delay (ps)	Sleep Tx Resistance (Ω)	$\sum R_i^{-1}$ (Ω^{-1})
CMOS	T_{M1}= 46 T_{M2}= 57	103	R_1=0 R_2=0	–
MTCMOS	T_{M1}= 50.6 T_{M2}= 62.7	113.3	R_1=250 R_2=9	0.1151
	T_{M1}= 52 T_{M2}= 61.3	113.3	R_1=330 R_2=2	0.5030
	T_{M1}= 48 T_{M2}= 65.3	113.3	R_1=110 R_2=25	0.0491

- 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

$$s'_n = \left[\min\left\{r'_{fanouts\ of\ C_n}\right\} - d'_n\right] - \left[\max\left\{a'_{fanins\ of\ C_n}\right\} + d'_n\right] \geq 0$$

slack node *n*

required time for node *n*

arrival time at node *n*

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

$$d'_n = d_n + \frac{v_i}{V_{DD} - V_{tL}} d_n$$

delay increase due to MTCMOS

- To simplify the constraints we only consider the timing critical paths ➔ need to define the notion of path delay!

Path Delay in MTCMOS

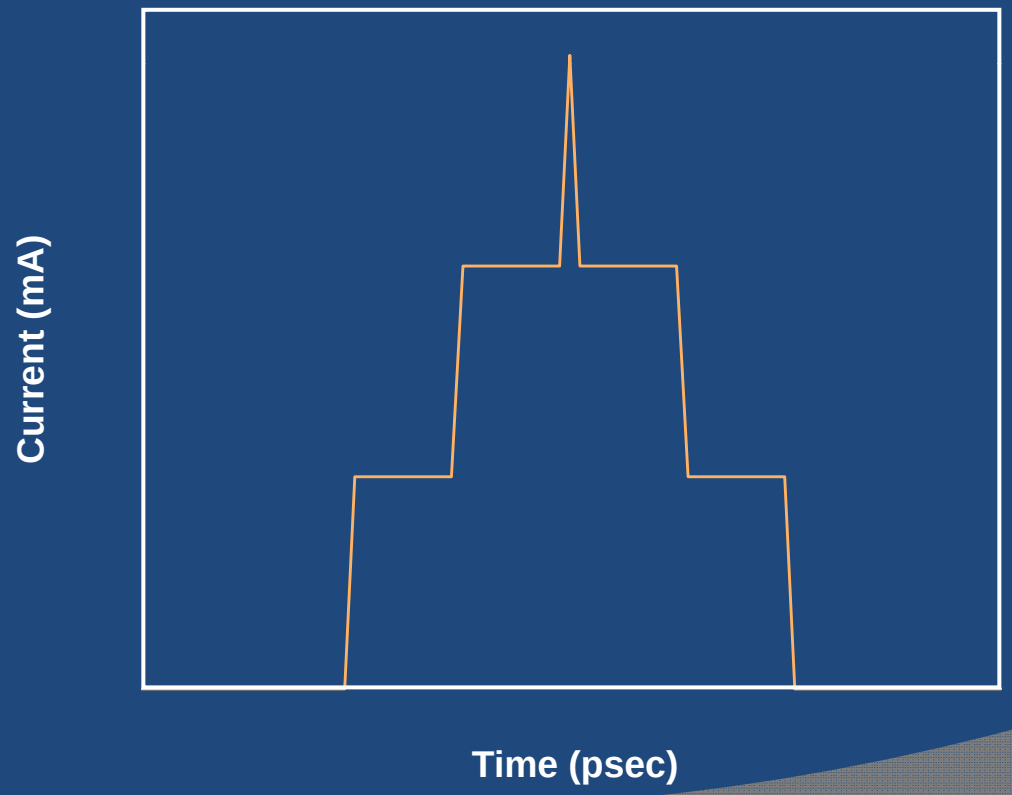
- The delay increase for path π_k is the summation of delay increases for all the gates in π_k:

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

  - $\theta(C_n)$ 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)
  - $I_{st_i}^{\max\left[t_{\min}^{C_n}, t_{\max}^{C_n}\right]}$ is the max current value flowing through R_{st_i} during the time window $\left[t_{\min}^{C_n}, t_{\max}^{C_n}\right]$ 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.

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

$$\text{s.t.}: \quad 1. \quad \Delta d_{\pi_k} = \sum_{C_n \in \pi_k} \frac{R_{st_i} I_{st_i}^{\max\left[t_{\min}^{C_n}, t_{\max}^{C_n}\right]}}{V_{DD} - V_{tL}} d_n \le \text{DDR\_MAX} \times d_{\max} \qquad 1 \le k \le K,\ 1 \le i \le N$$

$$2. \quad R_{st_i} I_{st_i}(t_j) \le \text{VVSS\_MAX};\ 1 \le i \le M, 1 \le j \le N$$

delay-budgeting constraints

static NM constraints

where:

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

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

BCM and MCM

- The delay-budgeting constraints can be written as:

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

- **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:

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

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:

$$LBCM = \underset{M_i = BCM}{\arg\min} \sum_{\substack{k=1 \\ \Delta d_{\pi_k} > \mathrm{DDR\_MAX} \times \mathrm{d_{max}}}}^{K} a_{ki}$$

- 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 for some i or j) **do**
7: M_m=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($R_{sti\text{-}initial}$, $I_{Mi}(t)$, 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 = ∞;
7: **for** *k=1* to *K*, *j=1* to *N*
8: **if** (Δd_{π_k} – DDR_MAX< min_slack)
9: min_slack = Δd_{π_k} – DDR_MAX×dmax;
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, α = 0.1

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

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