Sleep Transistor Sizing Using Timing Criticality and Temporal Currents

Anand Ramalingam*, Bin Zhang*, Anirudh Devgan† and David Z. Pan*
* Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712
† Austin Research Laboratory, IBM Research Division, Austin, TX 78758
{anandram,bzhang,dpan}@ece.utexas.edu and devgan@us.ibm.com

***Abstract*— Power gating is a circuit technique that enables high performance and low power operation. One of the challenges in power gating is sizing the sleep transistor which is used to gate the power supply. This paper presents a new methodology based on timing criticality and temporal currents to size the sleep transistor. The timing criticality information and temporal current estimation are obtained using static timing analyzer. The results obtained indicate that our proposed technique results in area reduction of sleep transistors by 80% and 49% compared to module based design and cluster based design respectively.**

I. Introduction

As technology scales, the supply voltage (V_{DD}) needs to be scaled down since it has a quadratic relationship with the dynamic power. But scaling down V_{DD} alone results in loss of performance. One way to maintain performance, is scaling down both V_{DD} and V_T [1]. But scaling down V_T exponentially increases the subthreshold leakage current. One of the techniques to reduce subthreshold leakage is power gating. Power gating is a circuit technique in which the source nodes of the gates in the functional block which were grounded are now connected to the drain of the NMOS sleep transistor. In the active mode, the sleep transistor is turned on to retain the functionality of the circuit. In the sleep mode, the sleep transistor is turned off, and the source nodes of the gates in the functional block float, thus cutting off the leakage path. Sleep transistor sizing is one of the major challenges in power gated circuits. If we overestimate the size we end up wasting silicon area and increasing the switching energy. If we underestimate the size, the required performance might not be achieved due to the increased resistance to the ground [2].

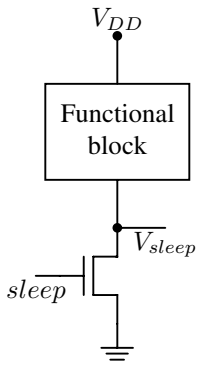

Fig. 1. Power gating

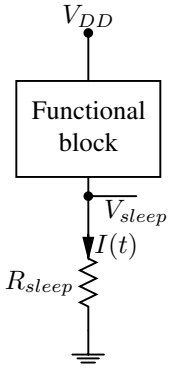

Fig. 2. Sleep transistor as a resistor

In the literature, various methods have been proposed to size the sleep transistor. In [3], *module* based design was proposed where a single sleep transistor is used for the entire circuit. In [4], the circuit is partitioned into *clusters* to minimize the maximum simultaneous switching current. Each cluster has an individual sleep transistor. In [5], all the individual sleep transistors of [4] are wired together and the resulting mesh is called the *distributed* sleep transistor network (DSTN). The discharging current is shared by the sleep transistor network which reduces the size of the sleep transistor. The sizing in all the above methodologies are based on the *maximum* worst case switching current I_{peak} [5].

In [2] it was shown that the sleep transistor can be approximated by a linear resistor that creates a finite voltage drop $V_{sleep} \approx R_{sleep}I(t)$ where $I(t)$ is the switching current through the sleep transistor as shown in Fig. 2. It is important to notice that different gates in a given path will see switching currents of *different magnitude* through the sleep transistor. The delay of a gate is inversely proportional to the gate drive $V_{GS} = V_{DD} - V_{sleep} = V_{DD} - R_{sleep}I(t)$ [6]. To the first order, we can state that the penalty experienced by each gate due to the sleep transistor is proportional to the current $I(t)$ flowing through the sleep transistor when the gate is switching. Hence if we are able to estimate $I(t)$ efficiently, we can use $I(t)$ to size the sleep transistor instead of I_{peak} which penalizes different gates in a path uniformly.

In this paper, we make the following contributions.

- Sleep transistor sizing making use of timing criticality and temporal switching current $I(t)$ of the circuit, and
- An efficient method to estimate the temporal switching current $I(t)$ of the circuit.

The results obtained indicate that our proposed technique results in area reduction of sleep transistors by 80% and 49% compared to module based design and cluster based design respectively.

The remaining part of the paper is organized as follows. Section II derives formula to size the sleep transistor. Section III presents a technique to estimate the switching current of a circuit and timing criticality based sleep transistor sizing is discussed in Section IV. Section V presents results for various benchmark circuits followed by conclusions in Section VI.

II. Sizing the sleep transistor

The delay of a gate (τ_d) can be expressed as [6],

$$\tau_d \propto \frac{C_L V_{DD}}{(V_{DD} - V_{T_L})^\alpha} \quad (1)$$

where C_L is the load capacitance at the gate output, V_{T_L} is the low threshold which is $0.7V$, $V_{DD} = 3.3V$, and the velocity saturation index $\alpha \approx 1$ for $0.18\mu m$ CMOS technology.

The delay of a gate with the sleep transistor can be expressed as,

$$\tau_d^{sleep} \propto \frac{C_L V_{DD}}{((V_{DD} - V_{sleep}) - V_{T_L})^\alpha} \quad (2)$$

where V_{sleep} is the potential of the virtual ground as shown in Fig. 1. Let $\tau_d^{sleep} = (1+\Delta)\tau_d$, where $\Delta\tau_d$ is the penalty due to the sleep transistor. Applying Taylor series to the denominator and approximating the sleep transistor as a linear resistor R_{sleep} [2], the penalty can be written as,

$$\Delta\tau_d \propto \frac{V_{sleep}}{V_{DD} - V_{T_L}}\tau_d = \frac{R_{sleep} I(t)}{V_{DD} - V_{T_L}}\tau_d \quad (3)$$

where $I(t)$ is the switching current through the sleep transistor.

A path in a circuit consists of various gates and these gates experience discharging currents of different magnitudes. From Equation 3, we find that the penalty for a gate due to the sleep transistor is proportional to $I(t)$. Now the delay penalty for a path ($\tau_{penalty}^{path}$) consisting of various gates can be written as,

$$\tau_{penalty}^{path} = \left(\frac{R_{sleep}}{V_{DD} - V_{T_L}}\right) \sum_{gate \in path} I_{local,max}\tau_d \quad (4)$$

where τ_d is the delay of the gate without the sleep transistor and $I_{local,max}$ is defined as,

$$I_{local,max} = max_{[t_1,t_2]} I(t) \quad (5)$$

where $[t_1, t_2]$ is the time interval over which the gate switches. Notice that the $I_{local,max}$ is the maximum *local* temporal current over the discharging timing window of the gate. We *differ* from the previous methodologies in this respect since they use maximum *global* current I_{peak}. Rearranging Equation 4,

$$R_{sleep} = \frac{(V_{DD} - V_{T_L})\,\tau_{penalty}^{path}}{\sum_{gate \in path} I_{local,max}\tau_d} \quad (6)$$

The current through the linearly-operating sleep transistor can be approximated as [4],

$$I_{sleep} \approx \mu_n C_{ox} \left(\frac{W}{L}\right)_{sleep} (V_{DD} - V_{T_L}) V_{sleep}$$

where μ_n is the mobility of electrons and C_{ox} is the oxide capacitance. Since the sleep transistor is operating in the linear region, then $R_{sleep} \approx \frac{V_{sleep}}{I_{sleep}}$. Then, the size of the sleep transistor can be written as,

$$\left(\frac{W}{L}\right)_{sleep} = \frac{1}{\mu_n C_{ox}(V_{DD} - V_{T_L})R_{sleep}} \quad (7)$$

Thus if R_{sleep} is known, the W_{sleep} can be determined directly. To determine R_{sleep}, we need an estimate of the temporal current flowing through the sleep transistor. The temporal current estimation technique is described next.

III. Temporal Current Estimation

We present a technique to estimate the worst case current discharged by a circuit. The current estimation technique needs timing windows of each gate and current expected to be discharged by each gate. The timing windows is obtained using $PrimeTime$ [7]. The expected discharge current (I_{exp}) of a gate is adapted from [4] and the pseudocode is shown below.

FIND-EXPECTED-CURRENT($gate$)
1 Find I_{peak} for each $gate$ in the library using HSPICE
2 $I_{exp} = \alpha_s \times I_{peak}$ ▷ α_s is the switching factor
3 **return** I_{exp}

The switching factor α_s is defined as the probability of the output (Y) switching. Thus α_s for falling output is,

$$\alpha_s = P\{Y = 1 \rightarrow 0 | Y = 1\} \times P\{Y = 1\}$$

We illustrate I_{exp} calculation using OR2. The switching factor $\alpha_s = \frac{1}{4} \times \frac{3}{4} = \frac{3}{16}$. From HSPICE simulations, we find that the $I_{peak} = 0.72ma$. Thus the expected current for an OR2 gate in our library is, $I_{exp} = \alpha_s \times I_{peak} = \frac{3}{16} \times 0.72ma = 0.12ma$.

After we have calculated I_{exp} for all the gates in our library we can use it for estimating switching current of a circuit and the pseudocode is presented below.

ESTIMATE-SWITCHING-CURRENT($circuit$)
1 Run $PrimeTime$ on the $circuit$ to get timing windows
2 $I(t) \leftarrow 0$
3 **for** every $gate$ in the $circuit$
4 **do** $I_{exp} \leftarrow$ GET-EXPECTED-CURRENT($gate$)
▷ Illustrated in Fig. 4
5 $I_{gate}(t) \leftarrow$ timing windows bounded by I_{exp}
6 $I(t) \leftarrow I(t) + I_{gate}(t)$ ▷ Illustrated in Fig. 5
7 **return** $I(t)$

We bound both falling and rising timing windows by the falling I_{exp}. The assumption is safe since for any gate, the worst case falling current through ground is always bigger than the short circuit current when the output rises. The switching factor α_s is the same for both falling and rising transitions.

To illustrate the current estimation procedure, consider the 1-bit carry lookahead adder (CLA) shown in Fig. 3. The timing analyzer $PrimeTime$ is run on this circuit to obtain the timing windows shown in Table I. Fig. 4 shows the currents associated with each timing window. To illustrate reading this graph, consider the OR2 gate O_1 in Fig. 3. The falling window for O_1 from Table I is $[73.92, 260.11]ps$. The I_{exp} of O_1 is $0.12ma$. Thus we have bounding rectangle of current $0.12ma$ over $[73.92, 260.11]ps$ as shown in Fig. 4. Finally, the currents across all the timing windows are summed up to find the total discharging current of 1-bit CLA shown in Fig. 5. Once the temporal switching current $I(t)$ has been estimated we can use that current to size the sleep transistor.

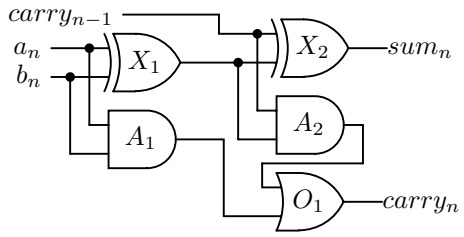

Fig. 3. 1-bit CLA

TABLE I
TIMING WINDOWS FOR 1-BIT CLA (TIME UNIT IS ps)

Gate	$rise_{min}$	$rise_{max}$	$fall_{min}$	$fall_{max}$
X_1	98.90	107.52	104.24	183.28
A_1	51.52	56.33	42.82	43.01
X_2	83.22	290.10	132.75	277.42
A_2	58.31	168.80	46.43	232.62
O_1	123.60	234.09	73.92	260.11

IV. TIMING CRITICALITY BASED SIZING

When a sleep transistor is inserted in a circuit, the performance of the circuit is penalized due to the reduction in driving voltage as evident in the Equation 2. In a macroscopic level, this translates to the fact that the paths are penalized. Thus if we are able to guarantee that the worst case path in the circuit, with sleep transistor switched on, satisfies the performance constraints then we can guarantee the performance of all the paths in the circuit.

There are two potential problems in sizing the sleep transistor based on paths. First, the number of paths in a circuit is exponential in size. Second, the worst case path for CMOS need not be the worst case path in MTCMOS [3]. To overcome the above two problems, we use a heuristic from static timing analysis (STA). The path based STA uses the top K worst paths to do optimization [7]. This idea is adapted to the sleep transistor sizing and the pseudocode is shown below.

SIZE-SLEEP-TRANSISTOR($circuit, K$)
1 Run $PrimeTime$ on the $circuit$ to get critical paths
2 $R_{sleep} \leftarrow \infty$
3 **for** $path \leftarrow 1$ **to** K ▷ Size using top K critical paths
4 **do** $R_{path} \leftarrow$ Size using Equation 6
5 $R_{sleep} \leftarrow$ MIN(R_{sleep}, R_{path})
6 $\left(\frac{W}{L}\right)_{sleep} \leftarrow$ Size using R_{sleep} in Equation 7
7 **return** $\left(\frac{W}{L}\right)_{sleep}$

To illustrate the sizing procedure, consider one of the worst case paths in the 1-bit CLA as shown in Table II. τ_d in Table II is the delay experienced by each gate without the sleep transistor. $I_{local,max}$ in Equation 6 differs for each gate in the path and it is got by looking up $I(t)$. For example, $I_{local,max}$ for X_2 is the maximum current discharged in the range $[183.28, 277.42]ps$. As shown in Fig. 5, the maximum current that flows in the above range is $I_{local,max}(X_2) = 0.92ma$. Note that we are using a *local* maximum to bound the current instead of the global maximum as in the previous methodologies. The above procedure is repeated for all gates in the path. Fig. 5 shows the local maximum currents seen by the gates X_1 and X_2 of the 1-bit CLA in Fig. 3. To illustrate the calculation of R_{path}, consider the path through X_1 and X_2. Let the penalty be 5% of the delay ($0.05 \times t_{arrival}$).

$$R_{path} = \frac{(3.3 - 0.7)\,(0.05 \times 277.42ps)}{183.28ps \times 1.3ma + 94.13ps \times 0.92ma} = 111\Omega$$

To illustrate the calculation of W_{sleep} we will assume the above R_{path} as the minimum resistance R_{sleep} obtained.

$$\left(\frac{W}{L}\right)_{sleep} = \frac{1}{1.25 \times 10^{-4}(3.3 - 0.7)111} = 27.77\lambda$$

Let $L_{sleep} = 2\lambda$ and we get $W_{sleep} = 55.55\lambda \approx 56\lambda$, where $\lambda = 0.1\mu m$.

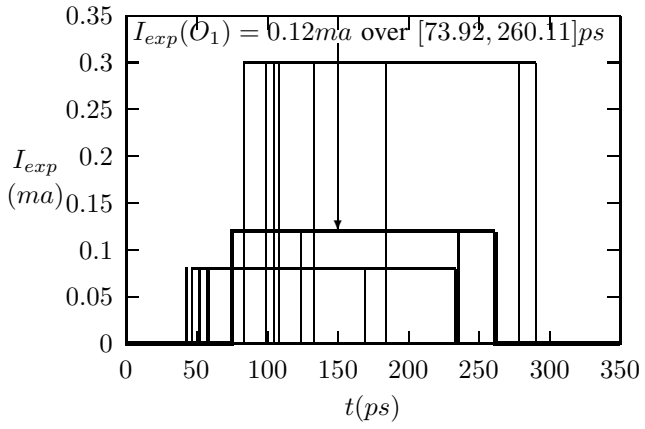

Fig. 4. I_{exp} bounding the *falling* and *rising* timing windows of each gate (Table I) in a 1-bit CLA. Refer to ESTIMATE-SWITCHING-CURRENT line 5

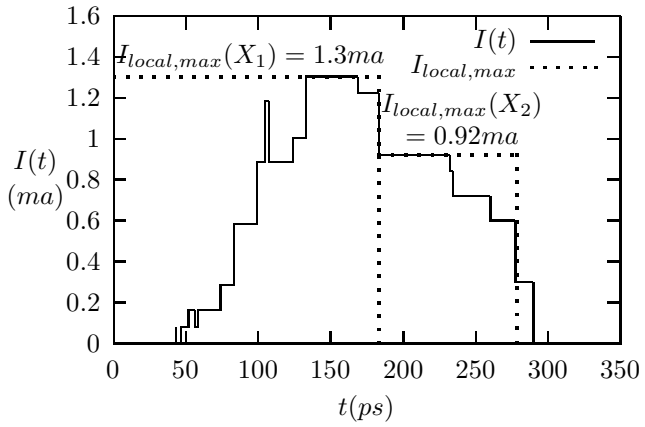

Fig. 5. The estimated current discharge $I(t)$ of a 1-bit CLA got by summing up all the currents in Fig. 4. Refer to ESTIMATE-SWITCHING-CURRENT line 6. Also shown are the local maximum currents seen by the gates X_1 and X_2. Refer to Equation 5. Note that by using local maximum instead of global maximum we reduce the size of the sleep transistor

TABLE II

A WORST CASE PATH IN 1-BIT CLA

Gate	$\tau_d(ps)$	$\tau_d^{path}(ps)$	fall/rise
X_1	183.28	183.28	fall
X_2	94.13	277.42	fall
$t_{arrival}$		277.42	

An important observation is that only the *falling inverting* gates are affected by the NMOS sleep transistor. Thus we penalize only the falling inverting gates in a path. Since the non-inverting gates in our library is a series combination of the inverting gate and the inverter, only the *rising non-inverting* gates are penalized.

V. RESULTS

The proposed sleep transistor sizing methodology has been implemented and its results are presented for various benchmark circuits. We use $0.18\mu m$ CMOS technology with $V_{DD} = 3.3V$, V_{T_L} = $0.7V$, and V_{T_H} = $0.9V$. L_{sleep} is set to $0.2\mu m$. The number of paths used to size the sleep transistor is set to $K = 100$ since $K > 100$ did not make any significant difference to the sizing.

In Table III under Module column, we compare our proposed module based sizing with module based sizing of [3]. We obtain an sleep transistor area improvement of 80% on

TABLE III

COMPARISON OF W_{sleep} OBTAINED USING MODULE AND CLUSTER BASED DESIGN FOR 5% PERFORMANCE DEGRADATION. THE UNIT IS $\lambda = 0.1\mu m$.

Circuit	Module (λ)		Cluster (λ)	
	[3]	Proposed	[4]	Proposed
CLA4	825	125	204	127
Parity checker	960	235	369	284
Wallace tree	1365	427	1201	698
c432	3438	475	1272	385
c499	3840	1171	2094	1351

an average over [3] since the proposed methodology has a global objective of satisfying performance for every path of the circuit while module based methodology has a restrictive local objective of satisfying performance for every gate. This coupled with the usage of $I(t)$ instead of I_{peak} leads to vast improvements in sizing.

In Table III under Cluster column, we compare our proposed cluster based sizing with cluster based sizing of [4]. We cluster such that the critical path is entirely within a single cluster. To discuss the results for clustering, we need to define *slack*. The slack in cluster c_j (S_{c_j}) for 5% performance penalty is defined as, $S_{c_j} = 1.05 \times CP_{circuit} - CP_{c_j}$, where $CP_{circuit}$ is the critical path in the entire circuit and CP_{c_j} is the critical path in cluster c_j. Suppose the entire circuit is divided into two clusters c_1 and c_2. Let c_2 contain the critical path which implies c_1 has more slack. This slack can be exploited to size the sleep transistor even smaller in c_1. Since we also size based on $I(t)$ instead of I_{peak} we obtain an sleep transistor area improvement of of 49% on an average over [4].

TABLE IV

COMPARISON OF W_{sleep} OBTAINED USING *proposed* METHODS FOR 5% PERFORMANCE DEGRADATION. THE UNIT IS $\lambda = 0.1\mu m$.

Circuit	Proposed (λ)	
	Module	Cluster
c880	638	509
c1908	479	457
c3540	1979	1933
c7552	12955	8325

In Table IV, we compare the sizes obtained using the proposed module and proposed cluster method. The circuits in Table IV have gate count in few thousand and have unbalanced paths. The presence of unbalanced paths is ideal for clustering as discussed earlier in regard to slack. The results validate our intuition that clustering is better for bigger circuits.

The results were verified with HSPICE simulations using random input vectors and also using the input vectors which exercise top K critical paths.

VI. CONCLUSIONS

We have introduced a new path based methodology to size sleep transistors using temporal currents and timing windows. We have also proposed an efficient method to estimate the temporal switching current $I(t)$ of the circuit. The results obtained indicate that our proposed technique results in area reduction of sleep transistors by 80% and 49% compared to module based design and cluster based design respectively.

ACKNOWLEDGMENT

Anand Ramalingam thanks Sreekumar V. Kodakara for his `perl` expertise. This work is partially sponsored by IBM Faculty Award. We used computers donated by Intel Corporation.

REFERENCES

[1] 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 Journal of Solid State Circuits*, 1995.

[2] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor sizing issues and tool for multi-threshold CMOS technology," in *Proceedings of Design Automation Conference*, 1997, pp. 409–414.

[3] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," in *Proceedings of Design Automation Conference*, 1998, pp. 495–500.

[4] M. Anis, S. Areibi, and M. Elmasry, "Design and optimization of multithreshold CMOS (MTCMOS) circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2003.

[5] C. Long and L. He, "Distributed sleep transistor network for power reduction," in *Proceedings of Design Automation Conference*, 2003, pp. 181–186.

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

[7] H. Bhatnagar, *Advanced ASIC Chip Synthesis: Using Synopsys Design Compiler, Physical Compiler and PrimeTime.* Kluwer Academic Publishers, 1999.