

# Timing Driven Power Gating

De-Shiuan Chiou\*, Shih-Hsin Chen\*, Shih-Chieh Chang\*, and Chingwei Yeh\*\*

\*Department of CS, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

\*\* Department of EE, National Chung Cheng University, Chiayi, Taiwan, R.O.C.

## ABSTRACT

Power Gating is effective for reducing leakage power. Previously, a Distributed Sleep Transistor Network (DSTN) was proposed to reduce the sleep transistor area by connecting all the virtual ground lines together to minimize the Maximum Instantaneous Current (MIC) through sleep transistors. In this paper, we propose a new methodology for determining the size of sleep transistors for the DSTN structure. We present novel algorithms and theorems for efficiently estimating a tight upper bound of the voltage drop. We also present efficient heuristics for minimizing the sizes of sleep transistors. Our experimental results are very exciting.

## Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids

## General Terms

Performance, Design

## Keywords

Leakage Current, Power Gating, IR Drop

## 1. INTRODUCTION

Sub-threshold leakage in standby mode is an important concern for many mobile designs that rely on low threshold devices to maintain operating speed under low supply voltages. One of the most effective ways to reduce the leakage is applying the multi-threshold-voltage CMOS (MTCMOS) technique. In the technique, a high  $V_t$  transistor, called the sleep transistor or power gate, is placed in series to the low  $V_t$  device as shown in Figure 1. The sleep transistor is turned off in standby mode to reduce leakage power, and turned on in active mode to maintain functionality while preserving timing constraints [7][9][10][11][12][13].

Three different ways of deploying sleep transistors have been proposed in the past. In the *module-based* design, a single sleep transistor is employed to support the power gating of the whole design [6][9]. In the *cluster-based* design in Figure 2(a), a circuit is decomposed into several clusters, and each cluster is connected to one *local* sleep transistor [1]. Finally, in the *distributed* design called *DSTN* in Figure 2(b), the cluster-based sleep transistor deployment is enhanced by connecting all the virtual ground lines

together, thus allowing the current from one cluster to flow through all sleep transistors [8]. In this way, the discharging current among the sleep transistors tends to be balanced. It has been demonstrated [8] that due to the balancing effect, DSTN consistently outperforms previous works.

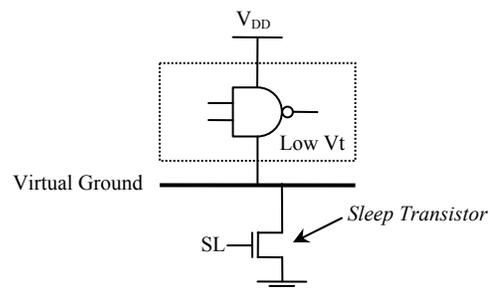


Figure 1: MTCMOS circuit scheme.

Nonetheless, DSTN greatly complicates a design decision that has always been one of the central themes of sleep transistor designs: transistor sizing. In its original form, the sizing of sleep transistors faces two opposing criteria. On the one hand, the leakage current through the sleep transistor is proportional to the size of the sleep transistor during standby mode, and hence, sleep transistors should be small enough to bound the leakage. On the other hand, the normal current flowing through sleep transistors during active mode will produce voltage drop that degrades the operating speed of the circuit. Since the voltage drop is *inversely* proportional to the size of the sleep transistors, sleep transistors should be large enough to bound the voltage drop and therefore to bound performance penalty.

The two opposing criteria were dealt with in previous works [1][6][8][9] as follows. First, the maximum instantaneous current (MIC) of the whole circuit and the MICs of clusters are obtained. The MIC's information and the prescribed voltage drop are used to determine the sizes of sleep transistors. Although more accurate bonding between the MIC and the voltage drop can always be obtained by extensive simulation using tools such as Nanosim, the procedure is too slow to be practical because of many time-consuming simulations.

We propose a new methodology to determine the size of sleep transistors for the DSTN structure and present algorithms and theorems for efficiently estimating a tight upper bound of the voltage drop. The tight upper bound can be used to assure the performance penalty of inserting sleep transistors, thus simplifying the sizing of the sleep transistors. Accordingly, efficient heuristics to minimize the size of sleep transistors are proposed. Experimental results show that our method not only can achieve the voltage constraints but also have small size of sleep transistors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

Copyright 2006 ACM 1-59593-381-6/06/0007...\$5.00.

The paper is organized as follows. Section 2 presents background knowledge and previous works. Sections 3 and 4 propose theorems and algorithms for voltage drop estimation, as well as the sizing method. Section 5 gives experimental results and section 6 concludes the paper.

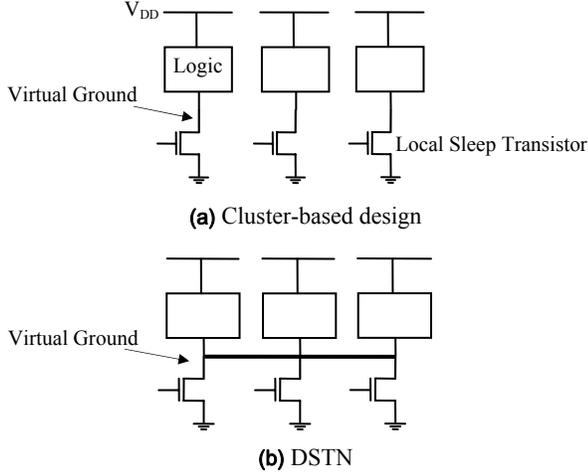


Figure 2: Different ways of deploying sleep transistors.

## 2. BACKGROUND

Since sleep transistors operate in linear region during active mode [5], the current through the sleep transistor can be expressed as:

$$I_{ST} \approx \mu_n C_{ox} \left( \frac{W_{ST}}{L} \right) (V_{DD} - V_t) V_{ST} \quad \text{EQ(1)}$$

where  $\mu_n$  is the N-mobility,  $C_{ox}$  is the oxide capacitance and  $V_t$  is the threshold voltage of the sleep transistor. Additionally, the variable  $V_{ST}$  is the source-drain voltage drop of a sleep transistor and is the voltage drop of the virtual ground line that degrades the circuit performance. After rewriting EQ(1), the voltage drop across a sleep transistor can be expressed as:

$$V_{ST} = k \left( \frac{I_{ST}}{W_{ST}} \right) \quad \text{EQ(2)}$$

where  $k = L/\mu_n C_{ox}(V_{DD} - V_t)$  can be treated as a constant. According to EQ(2), the voltage drop,  $V_{ST}$  is proportional to the current through the sleep transistor ( $I_{ST}$ ). However, the value of  $I_{ST}$  depends on the input vector applied to the circuit. If the maximum instantaneous current (MIC) through the sleep transistor,  $MIC(ST)$ , for all possible input vectors can be obtained, the maximum voltage drop can also be derived from EQ(2). As a result, given a pre-determined maximum voltage drop,  $V_{ST}^*$  and the value of  $MIC(ST)$ , the required width of sleep transistor width can be calculated as in EQ(3):

$$W_{ST}^* = k \left( \frac{MIC(ST)}{V_{ST}^*} \right) \quad \text{EQ(3)}$$

For the module-based design, the  $MIC(ST)$  is easy to obtain because  $MIC(ST)$  is equal to the MIC of the whole circuit. It is also true for the cluster-based design in Figure 2(a) as follows. Let sleep transistor  $ST_i$  be connected to cluster  $clus_i$ . The MIC of

$clus_i$ ,  $MIC(clus_i)$  is equal to  $MIC(ST_i)$  because each cluster is isolated. Therefore, the size of a sleep transistor should be proportional to the corresponding cluster's MIC.

Following the same argument, ideally, the size of a sleep transistor is also proportional to  $MIC(ST)$  for DSTN. However, due to the balancing effect of discharging currents in DSTN, it is difficult to estimate the MICs for all sleep transistors. In the previous work [8], weighted MIC of a circuit, i.e.,  $(1+\beta)*MIC(CKT)$ , are used with EQ(3) to calculate the ideal total area of all sleep transistors, where  $\beta$  is an empirical number between 0.05 and 0.5. The area is allocated to each sleep transistor in proportion to the corresponding cluster's MIC. Note that a small  $\beta$  leads to a large  $V_{ST}$ , which may violate the performance constraint, and a large  $\beta$  results in large area and leakage power penalty. However, there is no way to know the value of  $\beta$  except through trial-and-error. Moreover, since all virtual ground lines are tied together, using a lumped value of  $(1+\beta)*MIC(CKT)$  may fall short of the true current distribution among sleep transistors, thus compromising the quality of the final sizing.

## 3. FAST AND ACCURATE VOLTAGE DROP ESTIMATION

The description in the previous section demonstrates that a good estimation of  $MIC(ST_i)$  ascertains the worst case voltage drop in the virtual ground, and consequently contributes to a good sizing of sleep transistors. Hence, in this section we propose a method for deriving a tight upper bound for the  $MIC(ST_i)$ .

To begin with, both the sleep transistors and virtual ground lines are treated as resistors. Then, clusters are modeled as current sources dependent on input vectors. Therefore, DSTN is modeled as a resistance network in Figure 3, where  $R_{ST}$  is the resistance for a sleep transistors and  $R_V$  for a virtual ground lines.

Since the entire system is modeled as a linear system, the current flowing through a sleep transistor is the superposition of the currents from all sources to the designated sleep transistor. To see how the principle of superposition can be helpful in our method, consider one current source using the example in Figure 3. Let the resistances of sleep transistors be  $R_{ST} = (R_{ST1}, R_{ST2}, R_{ST3}, R_{ST4}) = (8, 9, 8, 10)$  and the resistance of virtual ground lines be  $R_V = (R_{V1}, R_{V2}, R_{V3}) = (1, 2, 2)$ . Let us assume the current source of the first cluster to be  $I_{clus1}$ . The currents along the sleep transistors contributed by  $I_{clus1}$  are  $\{0.38I_{clus1}, 0.27I_{clus1}, 0.21I_{clus1}, 0.14I_{clus1}\}$  according to Kirchhoff's Current Law and Ohm's Law.

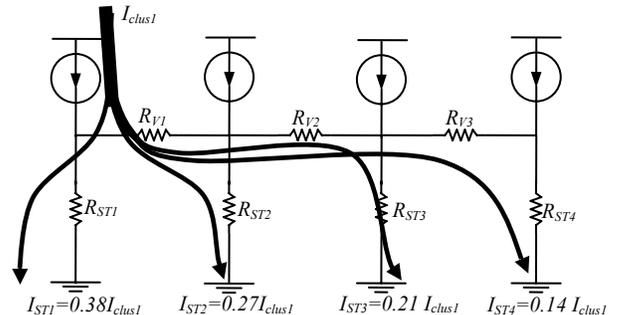


Figure 3: Current discharging with constant ratios.

Similarly, for each current source, we can obtain the current distribution along each sleep transistor. When the results of all current sources are superposed, we get the whole picture of current distribution, which can be mathematically written as

$$I_{ST} = \Phi \cdot I_{CLUS} :$$

$$\begin{bmatrix} I_{ST1} \\ I_{ST2} \\ I_{ST3} \\ I_{ST4} \end{bmatrix} = \begin{bmatrix} 0.38 & 0.30 & 0.21 & 0.18 \\ 0.27 & 0.30 & 0.21 & 0.18 \\ 0.21 & 0.24 & 0.35 & 0.28 \\ 0.14 & 0.16 & 0.23 & 0.36 \end{bmatrix} \begin{bmatrix} I_{clus1} \\ I_{clus2} \\ I_{clus3} \\ I_{clus4} \end{bmatrix} \quad \text{EQ(4)}$$

where  $\Phi = \{r_{ij}, 1 \leq i, j \leq n\}$  is the *Discharging Matrix*,  $r_{ij}$  is the amount of current flowing into  $ST_i$  when a unit current is drawn from a cluster  $clus_j$ , and  $n$  is the number of clusters.

Note that current  $I_{CLUS} = \{I_{clusi}\} = \{I_{clus1}, I_{clus2}, I_{clus3}, I_{clus4}\}$  discharging from clusters depends on the input vectors. In other words, for an input vector, we can obtain  $I_{CLUS}$ ; and then with  $I_{CLUS}$ , we can find  $I_{ST} = \{I_{STi}\} = \{I_{ST1}, I_{ST2}, I_{ST3}, I_{ST4}\} = \Phi \cdot I_{CLUS}$ . However, this is very time consuming for two reasons. First, for each single input vector we need to perform a matrix operations in EQ(4) once. Thus, to find maximum  $I_{STi}$  among all input vectors, many matrix operations are needed. Secondly, to achieve optimal sizing, we need to evaluate the performance penalties for different sizes of sleep transistors, with each requiring the estimation of the maximum  $I_{ST}$ .

Therefore, it is necessary to have a fast estimation of the maximum  $I_{STi}$ . In the following, we describe how to efficiently estimate a tight upper bound of the maximum  $I_{STi}$ . Similar to previous works [1][6][8][9], we utilize the MICs of clusters as constraints for  $I_{CLUS}$ . Note that finding the MICs of clusters is fast because it does not involve any matrix calculation, and there exist efficient heuristics [2][3][4] for this purpose. Thus, we assume the maximum instantaneous currents  $\{MIC(I_{clus1}), MIC(I_{clus2}), MIC(I_{clus3}), MIC(I_{clus4})\}$  for all clusters are known in advance. In fact, we can easily obtain many different combinations of MICs. For example,  $MIC(I_{clus1}, I_{clus2})$  is the maximum instantaneous current of a module composed of  $clus1$  and  $clus2$ . Due to the non-exclusive nature of MICs, it is always true that  $MIC(I_{clus1}, I_{clus2}) \leq MIC(I_{clus1}) + MIC(I_{clus2})$ .

With the MIC information, our approach considers one sleep transistor at a time and tries to estimate the maximum  $I_{STi}$  for the intended sleep transistor. Use Figure 3 again as an example. The problem can be modeled as the following linear program:

$$\max I_{ST3} = I_{clus1} \times r_{31} + I_{clus2} \times r_{32} + I_{clus3} \times r_{33} + I_{clus4} \times r_{34}$$

subject to:

$$I_{clus1} + I_{clus2} + I_{clus3} + I_{clus4} \leq MIC(I_{clus1}, I_{clus2}, I_{clus3}, I_{clus4})$$

$$I_{clus1} + I_{clus2} \leq MIC(I_{clus1}, I_{clus2})$$

$$I_{clus3} + I_{clus4} \leq MIC(I_{clus3}, I_{clus4})$$

$$I_{clusi} \leq MIC(I_{clusi}) \quad \text{for } i = 1, 2, 3, 4.$$

Though it seems easy to find the maximum  $I_{ST3}$  by solving the LP problem using the traditional Simplex method, the MIC constraints actually contain a unique property that leads to very efficient computation. Figure 5 shows a fast exact algorithm solving the maximizing problem of  $I_{STi}$  under the MIC constraints.

**Theorem 1:** The algorithm in Figure 4 obtains the maximum  $I_{STi}$  satisfying the MIC constraints.

**Proof:** Omitted.

Though the algorithm found the maximum  $I_{STi}$  under the MIC constraints, the results are in fact an upper bound estimation. The reasons are that the worst case must satisfy the MIC constraints but the worst case may not occur when there is MIC.

#### MIC(ST) UPPER BOUND ESTIMATION

1. Find the  $I_{clusi}$  with the largest corresponding  $r_i$ ;
2. Maximize the  $I_{clusi}$  found in step 1 under all the MIC constraints;
3. Substitute  $I_{clusi}$  in all equations with the maximum value  $I_{clusi}^*$ ;
4. If not all the  $I_{clusi}^*$  has been calculated, goto step 1;
5. Substitute all  $I_{clusi}$  in the objective function with  $I_{clusi}^*$  to get  $MIC(ST)$ ;
6. Return  $MIC(ST)$ .

**Figure 4: The exact algorithm solving the maximizing problem under MIC constraints.**

## 4. DSTN SIZING

We now discuss the heuristics for sizing sleep transistor based on  $I_{ST}$ . The overall algorithm is shown in Figure 5. Our heuristic starts with the smallest sleep transistor in the library. Then, we use the algorithm in Figure 4 to estimate  $MIC(ST)$ , which is the largest current flowing through sleep transistors. Then, the sleep transistor with the worst voltage drop is chosen for resizing according to EQ(3). Once a sleep transistor is resized, we update the discharging matrix  $\Phi$  and recalculate all the sleep transistors' MICs as well as the voltage drops. The process continues until all voltage drops of sleep transistors meet the given constraints. The convergence of the process is guaranteed through the following lemma.

**Lemma 1:** Suppose a sleep transistor is enlarged. The maximum currents of other sleep transistors obtained are always smaller than that before upsizing.

**Proof:** Omitted.

#### RESIZING HEURISTIC

1. Initialize sleep transistors size;
2. Calculate discharging matrix  $\Phi$ ;
3. Update the sleep transistor MICs and voltage drops;
4. If all voltage drops meet the constraints, goto step 6;
5. Resize the  $ST$  with the worst voltage drop, goto step 2;
6. Return size of all  $STs$ .

**Figure 5: Heuristic for sizing all the sleep transistors.**

## 5. EXPERIMENTAL RESULTS

To evaluate our method, we re-implemented the work of [8] with  $\beta = 0.5$  and  $0.05$  and performed on a set of ISCAS'85 benchmark circuits. In our experiment, we use TSMC 0.18um CMOS technology and  $V_{DD} = 1.8$  volt. A circuit is first synthesized into gate level netlists by Synopsys Design Vision. Then, the netlists

**Table 1: Total sleep transistor area.**

Circuit	#Cluster	#Gate	#PI	#PO	Area (Width) ( $\mu\text{m}$ )			Reduction (%)	Worst case voltage drop (volt)	Run time of our method (sec.)
					[8]		ours			
					$\beta=0.5$	$\beta=0.05$				
C432	17	323	36	7	61	43	41	32.8	-	0.9
C499	27	640	41	32	63	44	43	31.7	-	1.2
C880	23	528	60	26	63	44	42	33.3	-	1.9
C1355	29	625	41	32	95	67	64	32.6	-	2.2
C1908	27	830	33	25	88	62	60	31.8	-	2.2
C2670	35	1459	233	140	127	89	87	31.5	-	5.1
C3540	39	1613	50	22	192	135 failed	132	31.3	0.09054	6.6
C5315	49	2813	178	123	227	159 failed	163	28.2	0.0945	9.9
C6288	61	2464	32	32	1127	789 failed	864	23.3	0.10356	29.3
C7552	55	3685	207	108	376	263 failed	273	27.4	0.09882	15
<b>Avg.</b>					1.37	1.04	1	30.39		7.43

are placed, and routed by the Cadence Silicon Ensemble. After placement, the DEF file is exported to extract the physical location of each cell and the cells in the same row are grouped into a cluster. The MICs of a cluster is obtained via 10,000 PrimePower random simulations. The maximum tolerable voltage drop is set to be 0.09 volt. With MIC information and the maximum tolerable voltage drop, we obtain sizing of sleep transistors by [8] and our method. After that, sleep transistors are placed.

The experimental results are shown in Table 1. Columns 1, 2, 3, 4 and 5 give the name, the number of clusters, the number of gates, the number of primary inputs, and the number of primary outputs of a circuit, respectively. Columns 6, 7 and 8 show the area result of [8] with  $\beta = 0.5$ , [8] with  $\beta = 0.05$ , and our method. If the result does not meet the voltage constraint, we annotate the symbol “failed” in the right of the number. Column 9 shows the area reduction ratio of our method as compared to [8] with  $\beta = 0.5$ . Column 10 shows the worst case voltage drop of the four cases which fails the voltage constraint by [8] with  $\beta = 0.05$ . Column 11 shows the run time of our method.

The results show that our method uses smaller area than [8] with  $\beta = 0.5$ —as much as a 30% average reduction achieved. For [8] with  $\beta = 0.05$ , there are four circuits which cannot meet the voltage drop constraints. For circuits passing voltage constraints, our method has slightly less area than [8] with  $\beta = 0.05$  in average.

## 6. CONCLUSION

We revealed several useful properties concerning to the sizing of sleep transistors, which previously are determined by an empirical number  $\beta$  between 0.05 and 0.5. Based on these properties, we proposed an effective sizing method for the distributed sleep transistor network, and have illustrated the advantages of this method in terms of fast sizing computation, reduced sleep transistor area, and assured voltage drop across the sleep transistors compared to the use of  $\beta$ .

## 7. ACKNOWLEDGMENTS

The authors thank Hsin-Shih Wang and his team from Faraday Technology Corporation for the helpful discussion in these topics.

## 8. REFERENCES

[1] Anis, M., and et al., “Dynamic and Leakage Power Reduction in MTCMOS Circuits Using an Automated

Efficient Gate Clustering Technique,” *Proc. of the 39<sup>th</sup> DAC*, pp. 480-485, 2002.

- [2] Hsieh, C. T., Lin, J. C., and Chang, S. C., “A Vectorless Estimation of Maximum Instantaneous Current for Sequential Circuits,” *Proc. of ICCAD*, pp. 537-540, 2004.
- [3] Jiang, Y.M., and et al., “Estimation of Maximum Power and Instantaneous Current using a Genetic Algorithm,” *Proc. of the Custom Integrated Circuits Conf*, pp. 135-138, 1997.
- [4] Kristic, A., and Cheng, K. T., “Vector Generation for Maximum Instantaneous Current through Supply Lines for CMOS Circuits,” *Proc. of the 34<sup>th</sup> DAC*, pp. 383-388, 1997.
- [5] Kao, J., and et al., “Transistor Sizing Issues and Tool for Multi-threshold CMOS Technology,” *Proc. of the 34<sup>th</sup> DAC*, pp. 409-414, 1997.
- [6] Kao, J., Narendra, S., and Chandrakasan, A., “MTCMOS Hierarchical Sizing based on Mutual Exclusive Discharge Patterns,” *Proc. of the 35<sup>th</sup> DAC*, pp. 495-500, 1998.
- [7] Kao, J., and et al., “Subthreshold leakage modeling and reduction techniques,” *Proc. of ICCAD*, pp. 141-148, 2002.
- [8] Long, C., and He, L., “Distributed Sleep Transistor Network for Power Reduction,” *Proc. of the 40<sup>th</sup> DAC*, pp. 181-186, 2003.
- [9] Mutoh, S., and et al., “1-V Power Supply High-Speed Digital Circuit Technology with Multi-Threshold Voltage CMOS,” *IEEE JSSC*, vol. 30, no. 8, pp. 847-854, Aug. 1995.
- [10] Roy, K., Mukhopadhyay, S., and Mahmoodi-Meimand, H., “Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits,” *Proc. of the IEEE*, vol. 91, no. 2, pp. 305-327, Feb. 2003.
- [11] Sirichotiyakul, S., et al., “Stand-by Power Minimization through Simultaneous Threshold Voltage Selection and Circuit Sizing,” *Proc. of the 36<sup>th</sup> DAC*, pp.436-441, 1999.
- [12] Tschanz, J. W., and et al., “Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors,” *IEEE JSSC*, vol. 38, no. 11, pp. 1838-1845, Nov. 2003.
- [13] Wei, L., and et al., “Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications,” *IEEE Transactions of VLSI Systems*, vol. 7, no. 1, pp. 16-24, Mar. 1999.