Fine-Grained Sleep Transistor Sizing Algorithm for Leakage Power Minimization

De-Shiuan Chiou, Da-Cheng Juan, Yu-Ting Chen, and Shih-Chieh Chang
Department of CS, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
dawson@cad37.cs.nthu.edu.tw

ABSTRACT
Power gating is one of the most effective ways to reduce leakage power. In this paper, we introduce a new relationship among Maximum Instantaneous Current, IR drops and sleep transistor networks from a temporal viewpoint. Based on this relationship, we propose an algorithm to reduce the total sizes of sleep transistors in Distributed Sleep Transistor Network designs. On average, the proposed method can achieve 21% reduction in the sleep transistor size.

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
In CMOS technology, leakage increases exponentially and becomes a significant drain on total power consumption. The leakage power is expected to reach more than 50% of total power in the 65nm technology generation [3][10][11]. Power gating is a popular method for reducing leakage.

Many power gating structures have been developed to reduce leakage power. The authors of [6][9] proposed the module-based structure and a corresponding sizing algorithm. In the cluster-based structure [1], a module is decomposed into several clusters supported by corresponding sleep transistors. Recently, most industrial power-gating designs have adopted the Distributed Sleep Transistor Network (DSTN) implementation, which uses the current discharge balance to further reduce the total sizes of sleep transistors [8][12]. Furthermore, in [2] the authors proposed a sizing algorithm based on DSTN that guarantees the fulfillment of the IR drop constraint.

Figure 1 shows a DSTN design, which uses sleep transistors as switches to shut off leakage current. In the standby mode, leakage current through a sleep transistor is directly proportional to the sizes of sleep transistors [14]. In contrast, the IR drop across a sleep transistor in the active mode varies inversely with the size. Consequently, sizing a sleep transistor becomes a key concern for designers because it greatly affects the leakage reduction and performance in a power gating design. Conventionally, this dilemma scenario can be modeled as a size minimization problem under a designer-specified IR-drop constraint. The worst-case IR drop across a sleep transistor takes place when the corresponding Maximum Instantaneous Current (MIC) flows through it. We can calculate the minimum size of a sleep transistor based on both the MIC and the IR-drop constraint. Although the MIC of a circuit module or the entire design is easy to measure, predicting the MIC across a sleep transistor is a time-consuming task.

DAC 2007, June 4–8, 2007, San Diego, California, USA.

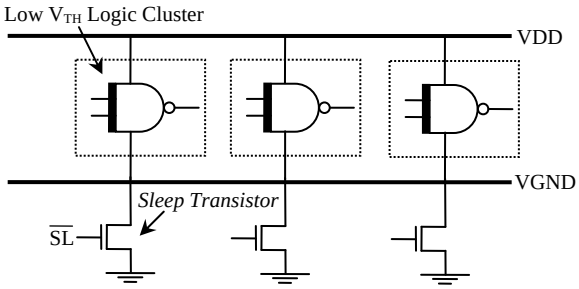


Figure 1. Power gating scheme.

We now describe an important feature of the MIC. Figure 2 shows the MIC distribution of two clusters of an industrial design where the MIC of each cluster occurs at different time points in a clock period. All the works [1][2][6][8][9] use the MIC of the clusters of the entire clock period to estimate the sleep transistor size. We observe that if a clock period is partitioned into several smaller time frames, the estimation of the MIC flowing through a sleep transistor will be more accurate. Nevertheless, optimization using many time frames leads to high computation complexity.

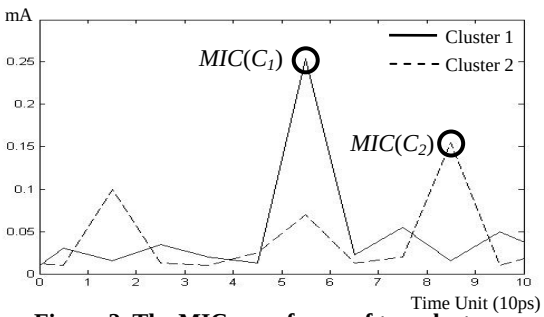


Figure 2. The MIC waveforms of two clusters.

In this paper, we derive an accurate MIC prediction across sleep transistors from cluster MICs in a temporal perspective for sleep transistor size minimization. We have also observed uniform time-frame partitioning is not efficient. Hence, we propose a variable length partitioning which significantly improves the runtime for computing an accurate IR-drop across a sleep transistor. A sizing algorithm using variable length time frames is also proposed. Our sizing method outperforms all previous works in regard to the total sizes of sleep transistors. Both the proposed partitioning and sizing methods can be successfully applied to [1][2][6][8][9] to improve the sizing results.

The remainder of this paper is organized as follows. In section 2, we introduce conventional methods for sizing mechanisms. In section 3, we perform an MIC analysis and present our time-frame partitioning approach and the sizing algorithm. Section 4 presents experimental results, and section 5 concludes the paper.

2. PRELIMINARIES

In this section, we introduce the prior arts for sleep transistor sizing. Suppose I_{ST} is the current flowing through the sleep transistor, and V_{ST} is the IR drop across the sleep transistor. The width W_{ST} can be calculated as:

$$W_{ST} = \left(\frac{I_{ST}}{V_{ST}} \right) \left(\frac{L}{\mu_n C_{ox} (V_{DD} - V_{TH})} \right) \qquad \text{EQ(1)}$$

The terms in the second parentheses are all constants, where L is the channel length, μ_n is the N-mobility, C_{ox} is the oxide capacitance, V_{DD} is the ideal supply voltage, and V_{TH} is the threshold voltage of the sleep transistor. Based on EQ(1), W_{ST} is proportional to I_{ST} under a designer-specified IR-drop constraint. Hence, when the largest I_{ST} is determined, the minimum required width W_{ST}^* can be described as:

$$W_{ST}^* = \left(\frac{MIC(ST)}{V_{ST}^*} \right) \cdot k \qquad \text{EQ(2)}$$

where V_{ST}^* is the IR-drop constraint, $MIC(ST)$ represents the MIC flowing through the sleep transistor, and k is the constant described in EQ(1).

Figure 3 shows a DSTN design with three clusters. Each cluster C_i is connected to the corresponding sleep transistor ST_i and to other sleep transistors by virtual ground. We define $MIC(C_i)$ as the MIC of cluster C_i and $MIC(ST_i)$ as the MIC flowing through sleep transistor ST_i. In general, a power gating design can be represented as a resistance network, which is a linear system as shown in Figure 4. Sleep transistors can be modeled as resistors since they operate in the linear region in the active mode [5]. The resistance value of sleep transistor ST_i is represented as $R(ST_i)$. Logic clusters are modeled as current sources whose values depend on input patterns. Each segment of virtual ground is also modeled as a resistor.

We would like to mention that $MIC(C_i)$ calculation has been studied extensively in previous works [4][7][13][15] and is assumed given in this paper. However, because of the current discharge balance phenomenon for DSTN shown in Figure 4, the exact $MIC(ST_i)$ cannot be calculated easily. Though $MIC(ST_i)$ may be obtained through extensive post-layout simulations, it becomes impractical with increasing sizes of designs.

It is known that the upper bound of $MIC(ST_i)$ can be estimated by using the information of $MIC(C_i)$. Taking Figure 3 and Figure 4 as an example, the relationship between $MIC(C_i)$ and estimated upper bound $MIC(ST_i)$ can be described as:

$$\begin{bmatrix} MIC(ST_1) \\ MIC(ST_2) \\ MIC(ST_3) \end{bmatrix} = \Psi \cdot \begin{bmatrix} MIC(C_1) \\ MIC(C_2) \\ MIC(C_3) \end{bmatrix} \qquad \text{EQ(3)}$$

$$\text{and} \quad \Psi = \begin{bmatrix} \psi_{11} & \psi_{12} & \psi_{13} \\ \psi_{21} & \psi_{22} & \psi_{23} \\ \psi_{31} & \psi_{32} & \psi_{33} \end{bmatrix}$$

where Ψ is a 3x3 matrix constructed from the resistance network. All the entries are positive and can be directly obtained via resistance values. Take the entry ψ_{11} for example:

$$\psi_{11} = \frac{(R_V + R(ST_3)) \times R(ST_2) + R_V \times (R_V + R(ST_2) + R(ST_2))}{(R_V + R(ST_3)) \times R(ST_2) + (R_V + R(ST_1)) \times (R_V + R(ST_2) + R(ST_2))}$$

After each $MIC(ST_i)$ has been calculated, the required size can be directly obtained from EQ(2). Note that in the remainder of this paper, since the exact $MIC(ST_i)$ is difficult to obtain, the notation $MIC(ST_i)$ stands for the **estimated upper bound** of the MIC flowing through sleep transistor ST_i based on EQ(3).

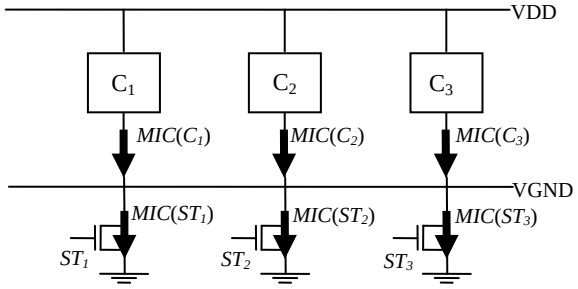


Figure 3. DSTN structure.

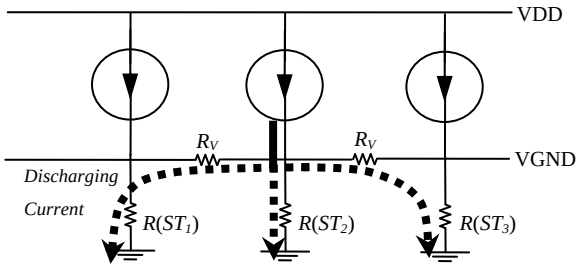


Figure 4. Resistance network modeling.

3. METHODOLOGY

In this section, we first explain why the concept of MIC temporal distribution leads to better sizing results. Then, we propose a variable length partitioning method to reduce the computational complexity, followed by a mathematical formulation of the sizing problem. Finally, we present the details of our sizing algorithm.

3.1 MIC Estimation Using Time-Frame Partitioning

In this section, we describe the concept of the MIC temporal distribution, and then explain how to use that concept to estimate *MIC*(*ST*) more accurately. The concept is illustrated in Figure 5. For the simplicity of the discussion, we consider the MIC waveforms of two clusters of an industrial Advanced Encryption Standard (AES) design. In Figure 5, $MIC(C_1)$ and $MIC(C_2)$ occur at different time points. Generally, the MICs of different clusters all behave in the same. We now describe how to improve the $MIC(ST_i)$ estimation using time-frame partitioning.

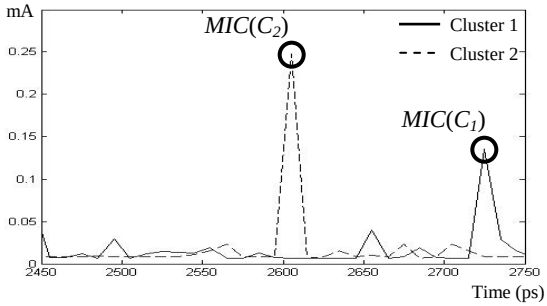


Figure 5. $MIC(C_i)$ waveforms of AES.

Given the waveform of $MIC(C_i)$, we partition one clock period uniformly into several time frames and collect the $MIC(C_i)$ of each time frame. The MIC of the i^{th} cluster, $MIC(C_i)$, is expanded into $MIC(C_i,T_j)$, which means the MIC of C_i in the j^{th} time frame. By definition, $MIC(C_i)$ is equal to the largest $MIC(C_i,T_j)$ for all j. We present this relationship in EQ(4).

$$MIC(C_i) = \max \{ MIC(C_i,T_j) \text{ for all } j \} \quad \text{EQ(4)}$$

Given the $MIC(C_i,T_j)$ information, we use EQ(3) to estimate $MIC(ST_i,T_j)$:

$$\begin{bmatrix} MIC(ST_1,T_j) \\ \vdots \\ MIC(ST_i,T_j) \end{bmatrix} = \Psi \cdot \begin{bmatrix} MIC(C_1,T_j) \\ \vdots \\ MIC(C_i,T_j) \end{bmatrix} \quad \text{EQ(5)}$$

where $MIC(ST_i,T_j)$ represents the MIC flowing through the i^{th} sleep transistor in the j^{th} time frame. Here we define $IMPR\_MIC(ST_i)$ as the largest value of $MIC(ST_i,T_j)$ for all j and present the relationship in EQ(6).

$$IMPR\_MIC(ST_i) = \max \{MIC(ST_i,T_j) \text{ for all } j \} \quad \text{EQ(6)}$$

In the following lemma, we explain that $IMPR\_MIC(ST_i)$ obtained from time-frame partitioning is more accurate than $MIC(ST_i)$ without partitioning.

Lemma 1: $MIC(ST_i) \geqq IMPR\_MIC(ST_i)$ for all i

Proof: From EQ(4) we have

$$MIC(C_i) \geqq MIC(C_i,T_j) \quad \text{for all } i, j \quad \text{EQ(7)}$$

By multiplying Ψ on both sides of EQ(7) we obtain EQ(8). Since the discharging matrix Ψ is a non-negative linear system, the inequality still holds.

$$MIC(ST_i) \geqq MIC(ST_i,T_j) \quad \text{for all } i, j \quad \text{EQ(8)}$$

Note that $MIC(ST_i)$ is the product of $MIC(C_i)$ and Ψ. Because $IMPR\_MIC(ST_i)$ is the largest value of $MIC(ST_i,T_j)$ for all j by definition, according to EQ(7), $MIC(ST_i)$ must be equal to or larger than $IMPR\_MIC(ST_i)$. □

For example, Figure 6 shows the relationship between $MIC(ST_i)$ and $IMPR\_MIC(ST_i)$. With $MIC(C_i)$ from Figure 5, we use EQ(3) to calculate the corresponding $MIC(ST_1)$ and $MIC(ST_2)$ shown as the horizontal dotted lines in Figure 6. Then we use EQ(5) and $MIC(C_i,T_j)$ from Figure 5 to calculate $MIC(ST_i,T_j)$ for all time frames. Figure 6 shows the waveforms of $MIC(ST_i,T_j)$. Furthermore, the marked two points are $IMPR\_MIC(ST_1)$ and $IMPR\_MIC(ST_2)$, which are 63% and 47% smaller than $MIC(ST_1)$ and $MIC(ST_2)$ respectively.

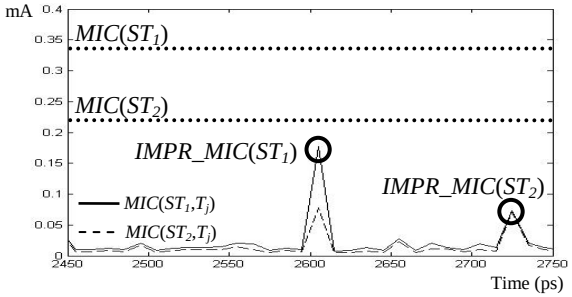


Figure 6. $MIC(ST_i,T_j)$ waveforms.

Since the worse-case IR drop across a sleep transistor is proportional to the MIC flowing through the sleep transistor, $IMPR\_MIC(ST_i)$ helps to predict the IR drop more accurately than $MIC(ST_i)$. We further derive **Lemma 2** from **Lemma 1** to conclude this section.

Lemma 2: The increasing number of time frames leads to smaller $IMPR\_MIC(ST_i)$ for all i.

Proof: Omitted.

3.2 Variable Length Time-Frame Partitioning

According to **Lemma 2**, having a large number of time frames leads to a smaller $IMPR\_MIC(ST_i)$. However, it also leads to high computation complexity. In this section, we use a technique of variable length time-frame partitioning to significantly reduce the complexity with only a slight loss in the accuracy of the $IMPR\_MIC(ST_i)$ estimation.

We first give a definition and a lemma.

Definition 1: For two different time frames T_a and T_b, T_a dominates T_b if $MIC(C_i,T_a) > MIC(C_i,T_b)$ for all i.

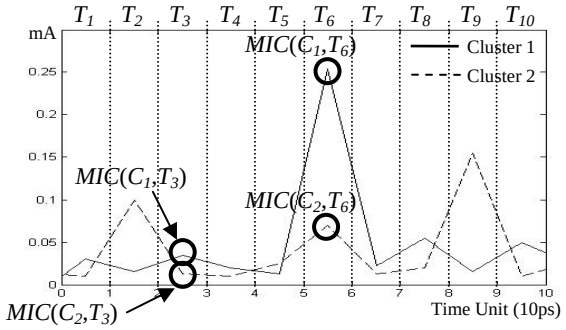
Lemma 3: If T_b is dominated by T_a, then $MIC(ST_i,T_a) > MIC(ST_i,T_b)$ for all i.

Proof: This can be directly proved by EQ(5).

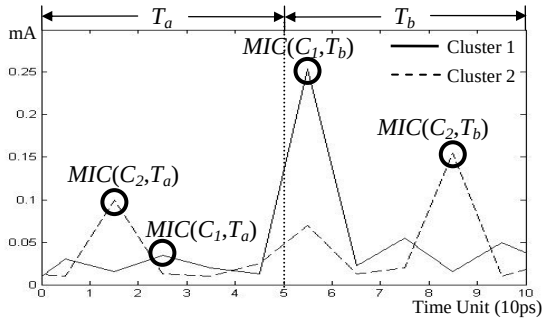
For example, Figure 7(a) shows the $MIC(C_i)$ distribution of a *uniform ten-way* partition. In Figure 7(a), time frame T_3 is dominated by T_6 because $MIC(C_1,T_6)$ is larger than $MIC(C_1,T_3)$ and $MIC(C_2,T_6)$ is larger than $MIC(C_2,T_3)$. From **Lemma 3**, we have that $MIC(ST_i,T_6)$ are larger than $MIC(ST_i,T_1)$ for all i. As a

result, we can neglect T_3 when calculating *IMPR_MIC*(ST_i). In this case, T_1, T_4, T_5, T_7, T_8, and T_{10} are also dominated by T_6. We can remove the dominated time frames to reduce the complexity.

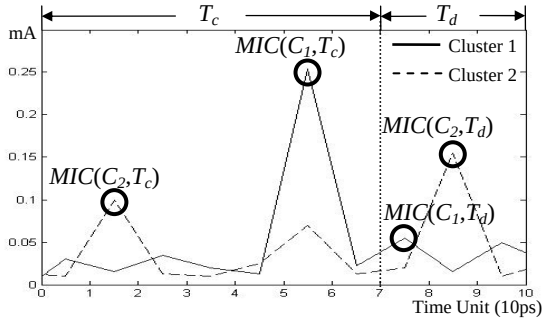
Now we discuss another important feature of the partitioning. Figures 7(b)(c) show two different ways of *two-way* partition. Figure 7(b) illustrates a *uniform two-way* partition while Figure 7(c) shows a variable length *two-way* partition. Since *MIC*(C_i,T_b) for all i in time frame T_b in Figure 7(b) are larger than *MIC*(C_i,T_c) and *MIC*(C_i,T_d) in Figure 7(c), T_c and T_d are dominated by T_b According to **Lemma 3** and EQ(6), *IMPR_MIC*(ST_i) in Figure 7(c) will be smaller than in Figure 7(b). This example shows that if all the *MIC*(C_i) are separated in different time frames, the *IMPR_MIC*(ST_i) can be better estimated.



(a) A *ten-way* partition.



(b) An inefficient *two-way* partition.



(c) An efficient *two-way* partition.

Figure 7. *MIC*(C_i,T_j) waveforms of different partitions.

The variable length time-frame partitioning problem can be formulated as follows. Given a specified number of time frame n, our objective is to find the efficient variable length *n-way* partition for *IMPR_MIC*(ST_i) estimation. Our algorithm works as follows. In the first step, we find the largest n+1 *MIC*(C_i) for all i. Secondly, we manage to partition these candidate values into different time frames. The exact partitioning point is in the middle of each two adjacent candidates. We reuse Figure 7(c) as an example. First, we search the time frames where an *MIC*(C_i) occurs. In this case, T_6 and T_9 are found. Secondly, assume n is 2, we mark the candidate time units and partition them into different time frames. The exact partitioning point is at 7. Figure 8 shows the details of our partitioning algorithm.

Our algorithm has a useful property: if n is smaller than the number of clusters, none of the variable length time frames obtained from Figure 8 is dominated by another. This can be directly proved from **Lemma 3**. Later in our experimental results, we always choose a small n less than the number of clusters, which still maintains the accuracy of the *IMPR_MIC*(ST_i) estimation.

Algorithm: *Time_Frame_Partitioning*(*MIC*(C_i,T_j), n)

```
1:  Output: An efficient partitioning
2:  /* step 1: mark the candidate time units */
3:  for j ← 1 to NUM_TF do
4:      if (the n+1 largest MIC(Ci) occurs in Tj) then
5:          mark(Tj);
6:      end if
7:  end for
8:  /* step 2: n-way partitioning */
9:  use n cuts to separate the marked Tj;
10: return
```

Figure 8. Variable length *n-way* partitioning algorithm.

3.3 Problem Formulation of ST Sizing

In this section, we formulate the problem regarding the sleep transistor size minimization on the DSTN structure under the given IR-drop constraint and the time frame information.

Inputs: 1. DROP_CONSTRAINT

2. *MIC*(C_i,T_j) for i = 1 to NUM_CLUSTER,
   j = 1 to NUM_TF.

Decision variables: $R(ST_i)$ $\forall i$

Objective function:

$$\text{minimize} \quad \sum_i \frac{\text{RW\_PRODUCT}}{R(ST_i)}$$

Subject to: $Slack(ST_i,T_j) \geqq 0 \ \forall i, j$

where $Slack(ST_i,T_j)$ =

DROP_CONSTRAINT- $MIC(ST_i,T_j) \times R(ST_i)$ EQ(9)

Figure 9. Sleep transistor sizing problem formulation.

The formulation of the sizing problem is presented in Figure 9. First, the time frame information and the IR-drop constraint (DROP_CONSTRAINT) are given as inputs of our sizing algorithm. Note that the time frame information includes the number of time frames (NUM_TF) and the corresponding *MIC*(C_i,T_j). The time-frame partitions can be uniform or variable length. The objective function is to minimize the total sleep transistor size. The voltage slack, $Slack(ST_i,T_j)$, is defined as the

drop constraint minus the product of $MIC(ST_i,T_j)$ and $R(ST_i)$. Therefore, if any slack is less than zero in a certain time frame, the IR-drop constraint is violated.

3.4 Sleep Transistor Sizing Algorithm

In this section we propose a sizing algorithm considering the properties of the time-frame partitioning. Based on **Lemma 1**, we can predict the IR drops more accurately from the $IMPR\_MIC(ST_i)$ estimation. With the accurate IR-drop prediction, we can calculate a set of $R(ST_i)$ that leads to a better sizing result. We observe that all $Slack(ST_i,T_j)$ can be enlarged as $R(ST_i)$ is scaled down. Hence, we initialize $R(ST_i)$ for all i with a large value. Then, we iteratively search the most negative $Slack(ST_i,T_j)$, and shrink corresponding $R(ST_i)$ until the constraint is satisfied. As a result, our method guarantees the IR-drop constraint.

The details of our sizing algorithm are presented in Figure 10. In the first step, all $R(ST_i)$ are initialized with a large value. Each time after the sizes are determined, we can obtain a new discharging matrix Ψ. With Ψ and $MIC(C_i,T_j)$, we are able to evaluate $MIC(ST_i,T_j)$ according to EQ(5). After that we use EQ(9) to update $Slack(ST_i,T_j)$ for each sleep transistors in all time frames. In the second step, we search the worst slack and enlarge the corresponding sleep transistor according to $MIC(ST_i,T_j)$. Since the size has been modified, consequently we can update Ψ, $MIC(ST_i,T_j)$, and the improved $Slack(ST_i,T_j)$ in the given order. Then, we run the second step iteratively until all slacks are equal to or greater than zero, which means the IR drop constraint is satisfied.

Algorithm: *ST_Sizing*($MIC(C_i,T_j)$, DROP_CONSTRAINT)

```
1:  Output: A set of decision variables R(ST_i)
2:  /* step 1: initialization */
3:  for i ← 1 to NUM_CLUSTER do
4:      R(ST_i) ← MAX;
5:  end for
6:  update Ψ, MIC(ST_i,T_j), and Slack(ST_i,T_j) for all i,j;
7:  /* step 2: sizing */
8:  repeat
9:      min_slack ← MAX;
10:     for i ← 1 to NUM_CLUSTER, j ← 1 to NUM_TF do
11:         if (Slack(ST_i,T_j) < min_slack) then
12:             min_slack ← Slack(ST_i,T_j);
13:             i* ← i;
14:             j* ← j;
15:         end if
16:     end for
17:     R(ST_i*) ← DROP_CONSTRAINT / MIC(ST_i*,T_j*);
18:     update Ψ, MIC(ST_i,T_j), and Slack(ST_i,T_j) for all i,j;
19: until Slack(ST_i,T_j) ≧ 0 for all i,j
20: return R(ST_i) for all i;
```

Figure 10. Sleep transistor sizing algorithm.

4. EXPERIMENTAL RESULTS

In our experiments, we implement the sizing methods of [2], [8], TP, and V-TP. TP represents our sizing algorithm with the clock period uniformly partitioned by the time unit (10 ps). V-TP represents our algorithm with the variable length *20-way* partition. All methods are applied to both MCNC benchmark circuits and an industrial AES design for comparison. The TSMC 130nm CMOS technology process is used in our experiments. Additionally, the virtual ground resistance is set to 0.057 ohm per micron according to the process data.

Figure 11 shows the sleep transistor sizing flow. An RTL netlist is synthesized by Synopsys Design Vision to output a gate-level netlist and an SDF file. For generating the VCD file, we use 10,000 random patterns to simulate the gate-level netlist with the SDF file. We partition the VCD file into several individual VCD files of each time frame. Meanwhile, the gate-level netlist is placed and routed by Cadence SOC Encounter. After placement, we obtain the DEF file to extract the location of each gate. The gates in the same row are grouped into a cluster. With the partitioned VCD files, we use PrimePower to evaluate the MIC of each cluster for all time frames. Note that the time interval of PrimePower is set to 10 ps, which is the time unit for cluster MIC measurement. The IR-drop constraint is set to 5% of the ideal supply voltage. With the MIC information and the IR-drop constraint, we apply the sizing algorithm of previous works and our own to obtain the experimental results.

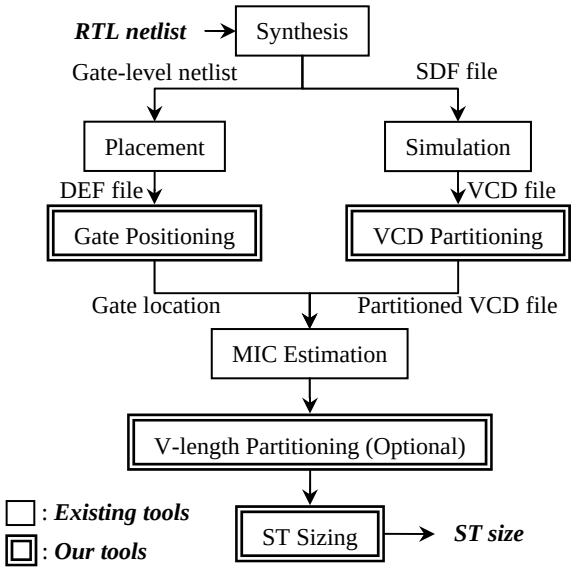


Figure 11: Implementation flow.

Table 1 shows the experimental results. Column 1 provides the name of benchmark circuits and Column 2 shows the gate counts. Columns 3, 4, 5 and 6 show the sleep transistor sizing results from [8], [2], TP and V-TP respectively. Take circuit t481 as an example. The sizing result is 9405 μm from [8], 7389 μm from [2], 5024 μm from TP, and 5402 μm from V-TP. The bottom row shows the average sizing results of all methods normalized to TP. On average, our TP method achieves 41% and 21% size reduction when compared to [8] and [2] respectively. The results clearly demonstrate that our method always achieves impressive size reduction on both benchmarks and the industrial design.

We also list the program runtime from TP and V-TP in Column 7 and 8. Results show that our variable length partitioning approach

greatly reduces 88% of runtime and only loses 5.6% sizing performance on average. Again, our improvement on the runtime is independent from the design complexity.

Table 1: Size and runtime results comparisons.

Circuit	Gate Counts	Total Area (Width in μm)				Runtime (Sec.)	
		[8]	[2]	TP	V-TP	TP	V-TP
C432	334	12817	8491	6775	7086	4262	495
C499	316	10741	8347	6684	7229	3644	568
C880	466	15050	11296	9233	9676	2561	345
C1355	339	19352	13056	10591	11496	2514	422
C1908	361	11859	8602	6988	7565	3173	273
C2670	295	5420	3183	2689	2756	1098	215
C3540	1010	29808	23020	18650	20282	16856	942
C5315	1248	29794	23773	18785	19534	13830	2190
C7552	1687	41016	29500	24269	25621	17216	2896
dalu	2395	3468	2904	2110	2283	3816	483
frg2	1712	3632	2835	2232	2255	701	136
i8	3781	13247	9931	7836	8141	7720	1080
t481	5316	9405	7389	5024	5402	16289	1514
des	6175	11804	9766	7850	8145	8321	1180
AES	40097	44378	33965	27229	28137	28379	3524
Avg.		**1.70**	**1.26**	**1**	**1.06**	**8.09**	**1**

Figure 12 shows part of the layout of the AES with sleep transistors inserted. Note that the AES is composed of 203 logic clusters and the design complexity is 40,097 gates. The sleep transistors are placed underneath the P/G network and the corresponding sizes are determined from our TP method.

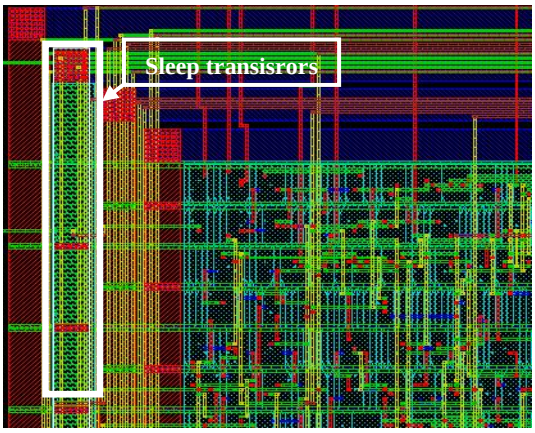


Figure 12: Sleep transistors in AES.

5. CONCLUSIONS

We have presented a new algorithm for reducing the leakage of a power gating design. The main idea of our method is to introduce the fine-grained $MIC(C_i)$ within a clock period from a temporal perspective. We also perform an intelligent time-frame partitioning for utilizing the given number of time frames. The results show that on average our algorithm can achieve 21% size reduction as well as leakage power reduction than [2].

6. REFERENCES

[1] M. Anis, S. Areibi, M. Mahmoud, and M. Elmasry , "Dynamic and Leakage Power Reduction in MTCMOS Circuits Using an Automated Efficient Gate Clustering Technique," *Proc. of the DAC*, pp. 480-485, 2002.

[2] D. S. Chiou, S. H. Chen, S. C. Chang, and C. Yeh, "Timing Driven Power Gating," *Proc. of the DAC*, pp. 121-124, 2006.

[3] H. Chang and S. S. Sapatnekar, "Full-Chip Analysis of Leakage Power Under Process Variations, Including Spatial Correlations," *Proc. of the DAC*, pp. 523-528, 2005.

[4] C. T. Hsieh, J. C. Lin, and S. C. Chang, "A Vectorless Estimation of Maximum Instantaneous Current for Sequential Circuits," *Proc. of ICCAD*, pp. 537-540, 2004.

[5] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor Sizing Issues and Tool for Multi-Threshold CMOS Technology," *Proc. of the DAC*, pp. 409-414, 1997.

[6] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS Hierarchical Sizing Based on Mutual Exclusive Discharging Patterns," *Proc. of the DAC*, pp. 495-500, 1998.

[7] H. Kriplani, F. Najm, and I. N. Hajj, "Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: algorithms, signal correlations, and their resolution," *IEEE Transaction on CAD*, vol. 14, no. 8, pp. 998-1012, Aug. 1995.

[8] C. Long, and L. He, "Distributed Sleep Transistor Network for Power Reduction," *IEEE Transaction on VLSI systems*, vol. 12, no. 9, Sep. 2004.

[9] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukuda, T. Kaneko, and J. Yamada, "A 1-V Multithreshold-Voltage CMOS Digital Signal Processor for Mobile Phone Application," *IEEE JSSC*, vol. 31, no. 11, Nov. 1996.

[10] R. R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Parametric Yield Estimation Considering Leakage Variability," *Proc. of the DAC*, pp. 442-447, 2004.

[11] K. Roy, S. Mukhopadhyay and H. M. Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proc. of the IEEE*, Feb. 2003.

[12] K. Shi, and D. Howard, "Challenges in Sleep Transistor Design and Implementation in Low-Power Designs," *Proc. of the DAC*, pp. 113-116, 2006.

[13] C. Y. Wang and K. Roy, "Maximization of power dissipation in large CMOS circuits considering spurious transitions," *IEEE Transaction on Circuits and Systems*, vol 47, no 4, pp. 483-490, Apr. 2000.

[14] W. Wang, M. Anis, and S. Areibi, "Fast Techniques for Standby Leakage Reduction in MTCMOS Circuits," *Proc. of the IEEE International SOCC*, pp. 21-24, 2004.

[15] Synopsys Inc. PrimePower Version-X 2005, 12 – User's Manual.