國立交通大學

電信工程學系

碩士論文

一個新穎且快速的調整睡眠電晶體尺寸演算法經由使用積分型的靈敏度

A Novel and Fast Sleep Transistor Sizing Algorithm by Using Integral Sensitivity

研 究 生：徐獻哲

指導教授：李育民　教授

中華民國九十八年一月

一個新穎且快速的調整睡眠電晶體尺寸演算法經由使用積分型的靈敏度
A Novel and Fast Sleep Transistor Sizing Algorithm by Using Integral Sensitivity

研究生：徐獻哲	Student：Shian-Je Shiu
指導教授：李育民	Advisor：Yu-Min Lee

國立交通大學
電信工程學系
碩士論文

A Thesis
Submitted to Department of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Communication Engineering

January 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年一月

一個新穎且快速的調整睡眠電晶體尺寸演算法經由使用積分型的靈敏度

學生：徐獻哲　　　　指導教授：李育民 博士

國立交通大學電信工程學系碩士班

摘　要

隨著製程進入深次微米時代，特別是在90奈米以下，因為互補式金屬氧化物半導體(CMOS)隨科技進步尺寸縮小，次臨界電流及閘漏電流呈指數增加，漏電功率已經變成主要的議題。所以，如何節省漏電功率變成很重要的議題。節省漏電功率的方法已經廣被發展，而電源閘是眾多有效方法的其中之一個。

在本篇論文裡，我們發展出一個新穎且快速的方法經由利用積分型的靈敏度去調整睡眠電晶體的尺寸。我們提出的演算法分為兩個階段。在第一個階段，我們利用改善卡洛夫子系統法(IEKS)解睡眠電晶體上的跨壓，在第二個階段則用正確的時域解取代改善卡洛夫子系統法(IEKS)。為了加速整個最佳化的過程，我們也發展出同時選取多個睡眠電晶體來調整。最後的實驗結果顯示我們的方法勝過目前已知最好的方法，它可以有效且快速的達到我們要的目標。

A Novel and Fast Sleep Transistor Sizing Algorithm by Using Integral Sensitivity

Student：Shian-Je Shiu   Advisor：Dr. Yu-Min Lee

Department of Communication Engineering
National Chiao Tung University

ABSTRACT

As the process technology enter the deep sub-micro era, especially below 90nm, leakage power has become a major issue due to the exponential increase of sub-threshold and gate leakage current with CMOS technology scaling. So, how to save leakage power becomes a very important issue. Many methods of save leakage power have been general developed, and power gating is one of the most effective methods.

In this thesis, we developed a novel and fast method by using integral sensitivity to size the width of sleep transistors. The purposed algorithm has two stages. In the first stage, we solve the voltage drop of sleep transistors by using Improved Extended Krylov Subspace (IEKS), and in the second stage, we employ exact time domain solver to replace IEKS. In order to speed up the optimizing procedure, a multiple sleep transistor simultaneously sizing strategy is also developed. The experimental results demonstrate that our method outperforms a state-of-the-art method, it can effective and fast to reach our object.

誌謝

這篇論文能順利完成，首先要感謝我的指導老師李育民教授。在我的研究生涯裡，每當遇到瓶頸，李老師總能給出許多寶貴的建議，引導我了解問題的根源，進而解決問題，讓我獲益良多。我相信這對我未來的成長是極有幫助的。另外在實驗室裡，李老師也提供一個非常棒的研究環境，不論是硬體或是軟體資源都極為豐富，所以實驗才能夠順利完成。

再者，要感謝至鴻學長、培育、柏毅，懷中在研究上給予的指點與建議。還要感謝實驗室的學弟妹，及昔日曾陪伴過我的伙伴。最後要感謝父母與哥哥，因為他們的支持與鼓勵，默默的陪伴我，讓我得以完成論文。最後由衷的感謝每位幫助我及關懷我的人，希望你們永遠健康快樂，謝謝你們。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

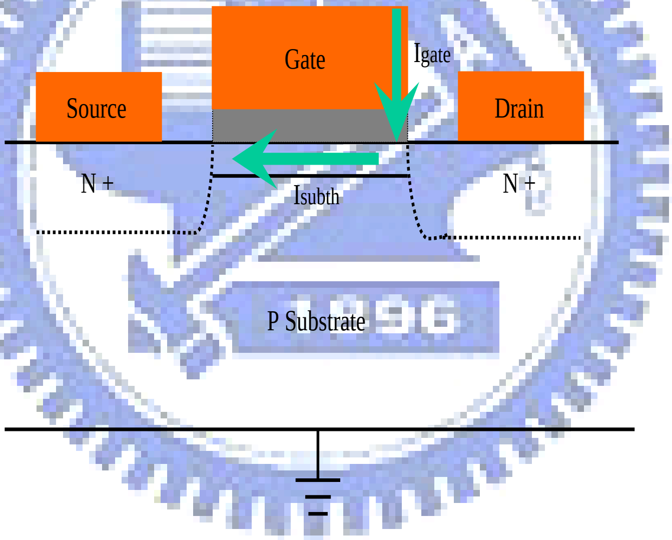


Fig. 1.1: Subthreshold leakage current and gate tunneling leakage current

As the process technology scales down to below $90nm$, leakage power has become a major issue. Two major leakage currents are the subthreshold leakage current and the gate tunneling leakage current, as shown in Fig. 1.1.

Subthreshold leakage current is the drain-to-source current of a transistor when the transistor is in the weak inversion region [1, 2]. Taking a NMOS transistor for example, when $V_{gs} < V_{th}$ and $V_{ds} = V_{dd}$, there is still a current flowing in the channel of the NMOS

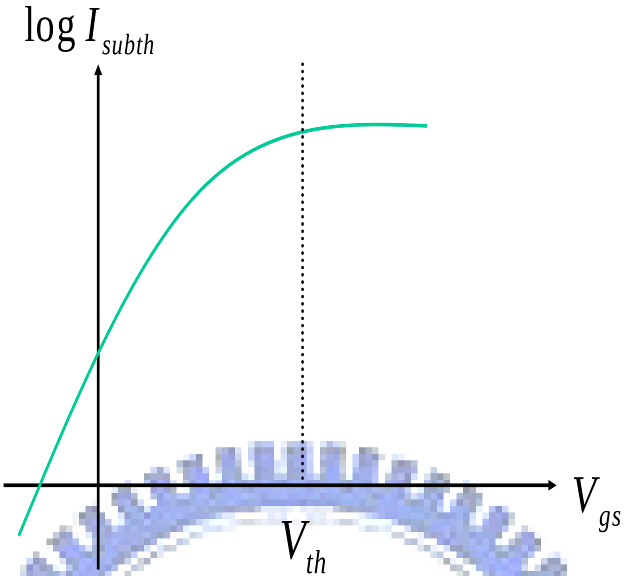


Fig. 1.2: The relation between I_{subth} and V_{gs}

transistor due to the V_{ds} potential. The subthreshold leakage of a MOS can be modeled as [3]

$$I_{subth} = A \times e^{1/mv_T(Vg-Vs-Vth_0-\gamma' \times Vs+\eta Vds)} \times (1 - e^{-Vds/V_T}), \tag{1.1}$$

$$A = u_0 C_{ox} \frac{W}{L_{eff}} (V_T)^2 e^{1.8} e^{-\Delta Vth/\eta V_T}, \tag{1.2}$$

where V_{th_0} is the zero bias threshold voltage, V_T is the thermal voltage, $\gamma\prime$ is the linearizeded body effect coefficient, C_{ox} is the gate oxide capacitance, μ_0 is the zero bias mobility, and m is the subthreshold swing coefficient of the transistor. In Equation (1.1), the V_{gs} and I_{subth} are exponentially related. Fig. 1.2 also illustrates the relation between V_{gs} and I_{subth} [2].

Gate tunneling leakage current is due to an electric field across the oxide coupled with a low oxide thickness, resulting in the tunneling of electrons from substrate to gate. The magnitude of the gate tunneling leakage current increases exponentially with the decrease of gate oxide thickness T_{ox} and the increase of supply voltage V_{dd}. In deep

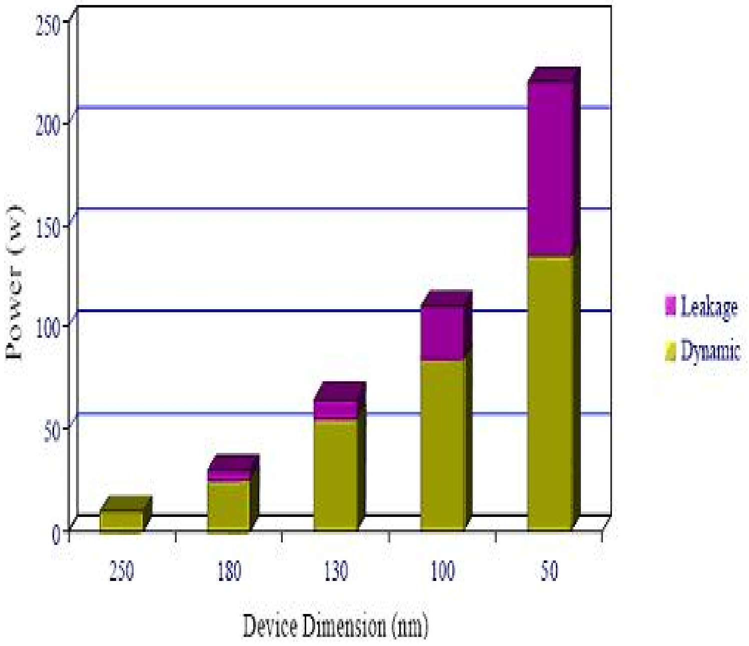


Fig. 1.3: Leakage power and dynamic power [5]

submicro VLSI design, the gate oxide thickness scales down as the process technology scales down. As the gate oxide thickness decreases, it results in increased gate tunneling leakage current. In order to reduce the gate tunneling leakage current, we can use a high-k dielectric. In addition to using a high-k dielectric, several additional leakage power reduction techniques have been developed such as MTCMOS (Multi-Threshold CMOS) and power gating. Power gating is also called sleeping transistor.

According to [4, 6], leakage power is expected to reach more than 50% of the total power of a chip in the 65nm technology. Furthermore, in ASIC system-on-chip VLSI design, leakage power is catching up with dynamic power, as shown in Fig. 1.3 [5]. Thus the issue of how to reduce leakage power is becoming a major current issue. Several useful methods have been proposed, as follows [1].

- **Power Gating:** When the circuit is in the standby state, the power gating is in the off state. Hence, the circuit's leakage current is blocked by the power gating and leakage power is reduced. Otherwise, as the circuit is in the active state, and the power gating is in the on state, it results in IR drop across the power gating. Hence, the performance of circuit is degraded.

- **Multi-Threshold CMOS (MTCMOS):** MTCMOS is a technology that uses several types of transistors with different threshold voltage values. Sleep transistors have high threshold voltage and transistors in the cluster have low threshold voltage. As we increase the threshold voltage of the sleep transistors, it reduces the diffusion of minority carriers, hence the leakage power is reduced.

- **Body Bias:** When the circuit is in the standby state, reverse-body voltage is applied to transistors so as to increase the threshold voltage, and then leakage current is reduced. The threshold voltage of a transistor can be calculated as

  $$V_T = V_{T0} + \gamma(\sqrt{|-2\Phi_F + V_{SB}|} - \sqrt{|-2\Phi_F|}), \tag{1.3}$$

  where V_{T0} is the threshold voltage for $V_{SB} = 0$, γ is the body-effect coefficient and Φ_F is the substrate Fermi potential. According to Equation (1.3), it can be observed that a reverse bias will increase a transistor's threshold voltage, thus reducing leakage current.

This thesis focuses on sleep transistors because they are a useful way to reduce leakage power. Sleep transistors are an effective method for reducing leakage power. However, sleep transistors have the drawback of increasing timing delay and the total area of a circuit. Accordingly, it is desired to minimize the total width of the sleep transistors subject to meet other circuit constraints, which generates a trade off question. The question can be written as a problem formulation, which along with our method will be presented in chapter 3.

1.2 Our Contributions

In this thesis we propose a novel and fast sleep transistor sizing algorithm. The sizing algorithm has two stages. In the first stage, Improved Extended Krylov subspace (IEKS) method is utilized to obtain the voltage of the nodes [7, 8], which is a very effective way to speed up the circuit analysis. We can get a better solution for the width of each sleep transistor after finishing the first stage, and then enter the second stage. In the second stage, the exact time domain solver is employed. It is a more accurate solver, it provides the best sleep transistor candidates for each sizing step and avoids oversizing. During the sizing procedure we consider the global region in the time period, therefore introducing both cost and integral sensitivity and also utilizing them to size the sleep transistors. The experimental results will show that the performance of our method is better than the state-of-the-art method [4].

1.3 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 introduces the basic concepts of sleep transistors and reviews the previous works. Chapter 3 presents flowchart and problem formulation, definition of cost and integral sensitivity, model order reduction techniques, the moment of unit pulse waveform, the sizing algorithm and speed up procedures. Experimental results and conclusions are presented in chapters 4 and 5, respectively.

Chapter 2

Background

In this chapter we first provide a basic background for power gating and several power gating structures. After that we survey the previous work.

2.1 Basic Concepts of Power Gating

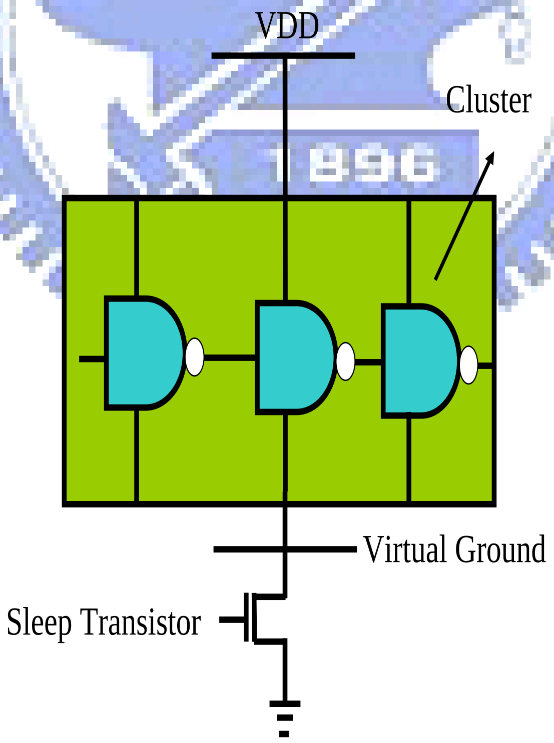


Fig. 2.1: A single sleep transistor

In modern VLSI circuit design, power gating is widely used to reduce leakage power.

Power gating also called sleep transistor . Generally, a sleep transistor is connected between a virtual ground and a ground, as shown in Fig. 2.1. When the logic or clusters are in the standby mode, the sleep transistor is in the off state, thus blocking the leakage current of the logic or clusters. When the logic or clusters are in the active mode, the sleep transistor is in the on state so there is a current through the sleep transistor, and resulting in an IR drop across the sleep transistor. The IR drop across the sleep transistor increases the timing delay of circuit, hence the performance of circuit is degraded.

2.2 Basic Formulations of Power Gating

A sleep transistor operates in the linear region when it is in the on state. Because the voltage drop between gate and drain is bigger than threshold voltage, so a sleep transistor can be modeled as a resistance [13, 14]. In the linear region, the current through a sleep transistor can be formulated as

$$I_i^{st} \approx \mu_n C_{ox}(\frac{W_i^{st}}{L})(V_{DD} - V_{th})V_i^{st}, \tag{2.1}$$

where μ_n is the N-mobility, C_{ox} is the gate oxide capacitance, V_{th} is the threshold voltage of a sleep transistor, V_i^{st} is the voltage across the drain node and source node of the sleep transistor i, and W_i^{st} is the width of the sleep transistor i. For each sleep transistor, Equation (2.1) can be reformulated as

$$W_i^{st} = \frac{1}{k}(\frac{I_i^{st}}{V_i^{st}}), \tag{2.2}$$

where $k = \mu_n C_{ox}(V_{DD} - V_{th})/L$ can be treated as a constant. In the DSTN design, we wish the width of sleep transistor to be as small as possible, but if it is not large enough, the voltage drop across the sleep transistor will be larger than the maximum allowed voltage drop, and the discharge/charge current of the gate will be reduced. Hence, circuit performance will be degraded. This is a trade-off question between the sleep transistor width and the circuit delay. In this thesis, we propose a novel method that not only minimizes the total width of sleep transistors but also meets the maximum allowed voltage drop constraint. The details will be described in chapter 3.

2.3 Structures of Sleep Transistor

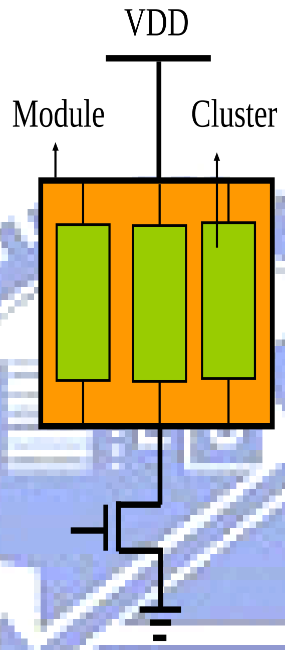


Fig. 2.2: Module based design

In this session we introduce three different structures, namely module based design, cluster based design and distributed sleep transistor network(DSTN) design. In the module based design, a module includes several clusters or logic, and the ends of the clusters are connected by a sleep transistor as shown in Fig. 2.2. The sleep transistors in the module based design are controlled by a power management processor (PMP), as shown in Fig. 2.3 [9]. The general design that can be traced back to [10, 11]. In the cluster based design, a cluster includes several cells, the end of the cluster is connected to a sleep transistor as shown in Fig. 2.4. Cluster based design can be found in [12]. In the distributed sleep transistor network (DSTN) design, a sleep transistor is inserted under a cluster or a

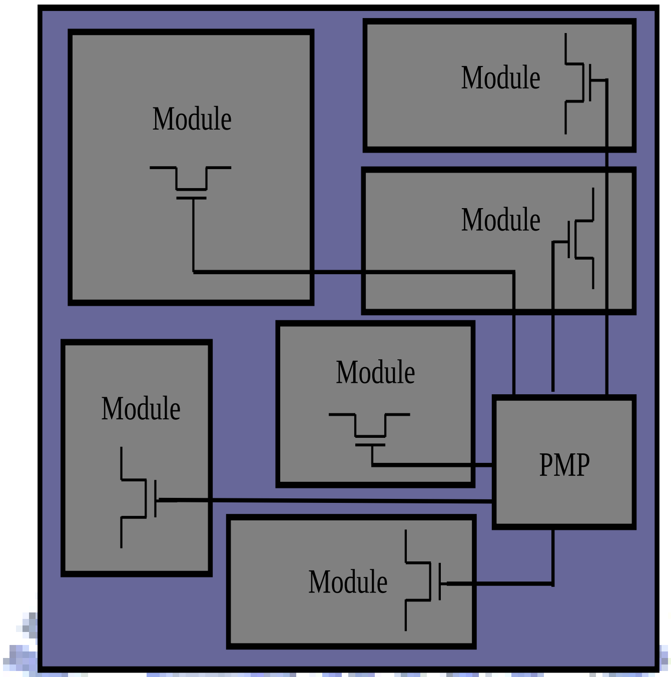


Fig. 2.3: Power management processor

cell, and the drains of these sleep transistors are connected together as shown in Fig. 2.5. By comparing the module based design with the cluster based design, we can observe the maximum instantaneous current (MIC) of the module is actually less than summation MIC of the clusters in the module, so the total area of module based design is less than cluster based design if we do not consider the resistances of the virtual ground wires. However, the long virtual ground wires in the module based design increase the IR drop, which may result in functional failure. In order to avoid this situation, it is necessary to increase the width of the sleep transistors. On the other hand, in the DSTN design, since the MIC of each cluster usually does not occur in the same time and all the sleep transistors share all the currents of the clusters, therefore the total area of the sleep transistors is reduced. The previous works also show that the DSTN design is better than cluster based design and module based design. Hence, we employ the DSTN design to reduce leakage current in this thesis.

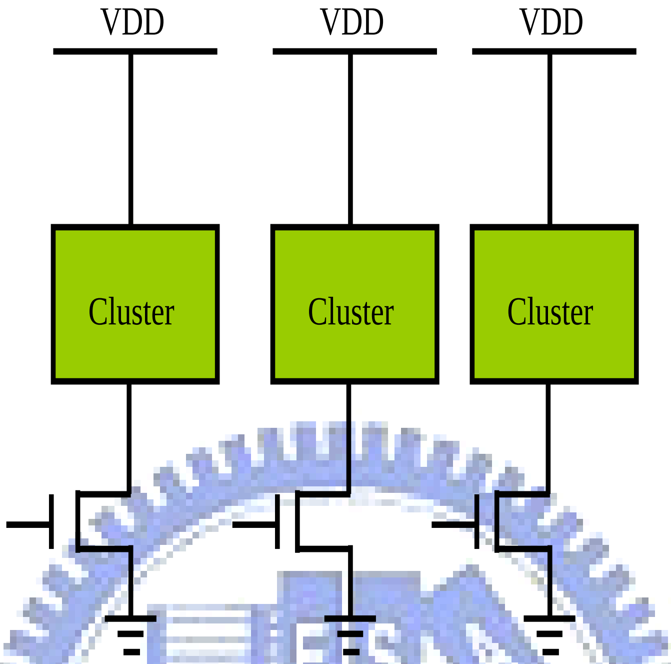


Fig. 2.4: Cluster based design

2.4 Previous Works

In this section we review the previous work. In [11], module based design was proposed so as to use a single sleep transistor for the entire circuit. In [15], cluster based design was presented, whereby each cluster has an individual sleep transistor. In [9], the authors proposed a novel distributed sleep transistor network (DSTN) such that each cell or cluster was connected to an individual sleep transistor and the drains of the sleep transistors were connected together. The experimental results showed that the DSTN design reduced the area better than the cluster based design. In [9], the authors estimated the total width of the sleep transistors by Equation (2.3), where $MIC(CKT)$ is the maximum instantaneous current of the circuit and β is an empirical number to consider the effect of the resistances of the virtual ground lines. β can be calculated by Equation (2.4), where N_{clu} is the

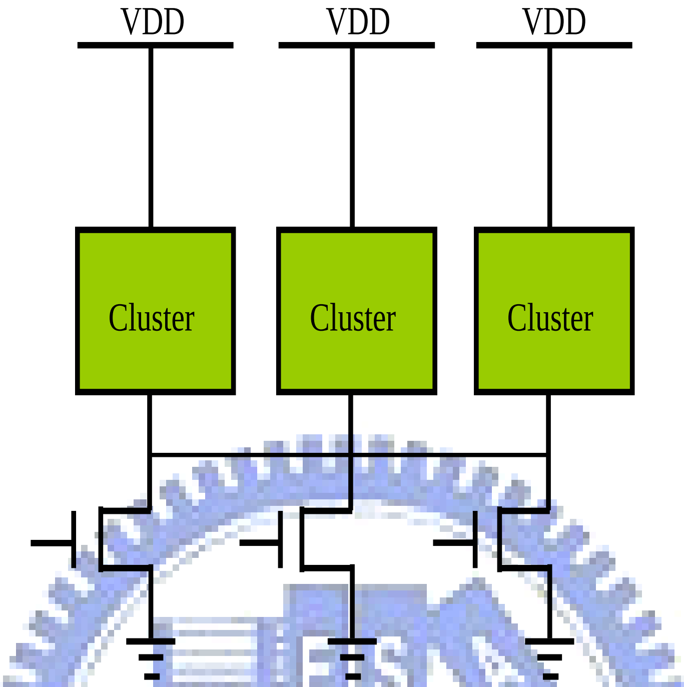


Fig. 2.5: Distributed sleep transistor network (DSTN)

number of clusters in the circuit.

$$W^{st} = \frac{1}{k}(\frac{(1+\beta)*MIC(CKT)}{V^{st*}}), \tag{2.3}$$

$$\beta = 0.002 * N_{clu}. \tag{2.4}$$

In [16], the authors presented a sleep transistor sizing algorithm which considered timing criticality and temporal currents. In [14], the authors utilized the DSTN structure and estimated a tight upper bound of the voltage drop. Recently, the authors in [4] partitioned the waveform of each cluster into many time frames and then estimated the MIC flowing through a sleep transistor at each time frame. Although the MIC can be accurately computed for the sizing step, it is time-consuming. Therefore the authors proposed a variable length partitioning scheme to speed up the procedure, but the total width of the sleep transistors was increased. In [4], a state-of-the-art method was proposed. The authors just only consider the maximum instantaneous voltage violation, and reduces the violation.

However, our method consider a time period for each node. The different between us is we consider the global region, [4] just only consider the local region. Finally, the results show that our method outperform the state-of-the-art method, it will be shown in chapter 4.

Chapter 3

Sleep Transistor Sizing Algorithm

In this chapter we present algorithm flowchart and problem formulation, DSTN modeling, definition and calculation of cost and integral sensitivity, model order reduction techniques and the details of the two-stage sizing algorithm.

3.1 The Flowchart and Problem Formulation

In this these, we proposed the sleep transistor sizing algorithm, the algorithm is also called two-stage sizing algorithm. Before the beginning, we firstly define the drain of sleep transistor i as node i. Fig. 3.1 illustrates the voltage waveform at node i, where V_i^{st*} is the maximum allowed threshold voltage at node i, V^{up} is the upper bound voltage at node i, $V_i^{max\_vio}$ is the maximum instantaneous voltage violation at node i, and the yellow parts of this figure are the cost of node i. The upper bound voltage $V^{up}$is employed in the first stage, it will help us to speed up the sizing procedure and get the initial solution for second stage.

Generally, the execution time is proportional to the number of independent sources. when the circuit become large, the runtime will be increase. In order to handle large circuit design, two-stage sizing algorithm is proposed. The first stage provides a initial solution for second stage, and the upper bound voltage and model order reduction method are employed to speed up the all of the sizing procedure. In the second stage, the exact time domain solver is utilized, it provides the final solution and make sure results are not

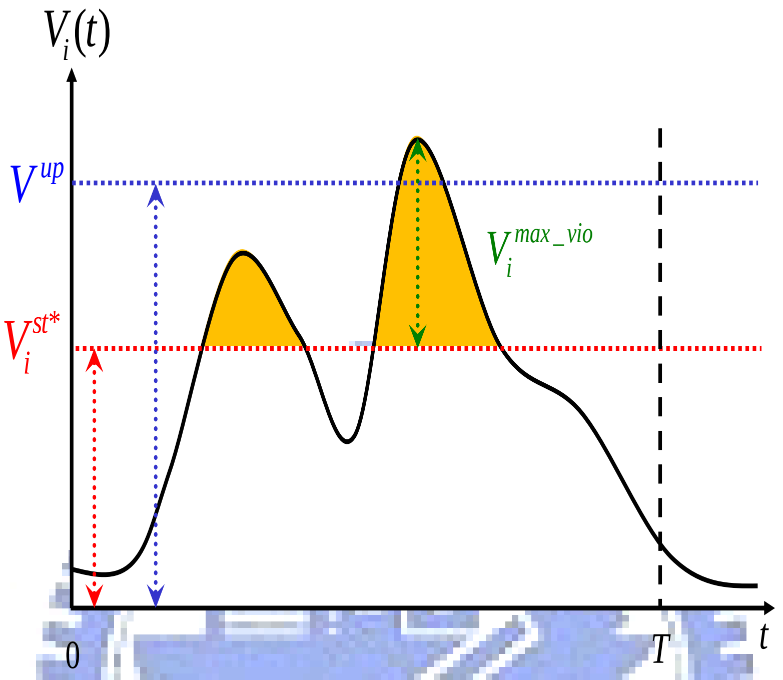


Fig. 3.1: The voltage waveform at node i.

violation circuit design. The proposed two-stage sizing algorithm design flow is shown in Fig. 3.2. The design flow has two stages. In the first stage, we utilize model order reduction to solve the voltage at total nodes and then construct the sizing metric. Next we size the sleep transistors and update the conductance matrix G, repeating the above procedure until it meets the upper bound constraint. In the second stage, we employ an exact time domain solver to replace the model order reduction solver, again repeating the procedure until it meets the maximum allowed threshold voltage constraint, i.e. meets the cost constraint. The details will be described in the following sections. Our problem formulation is as follows:

- **Input:** Given a set of independent piece wise linear(PWL) current sources, maximum allowed threshold voltage at each node, the upper bound voltage, and minimum width of sleep transistors.

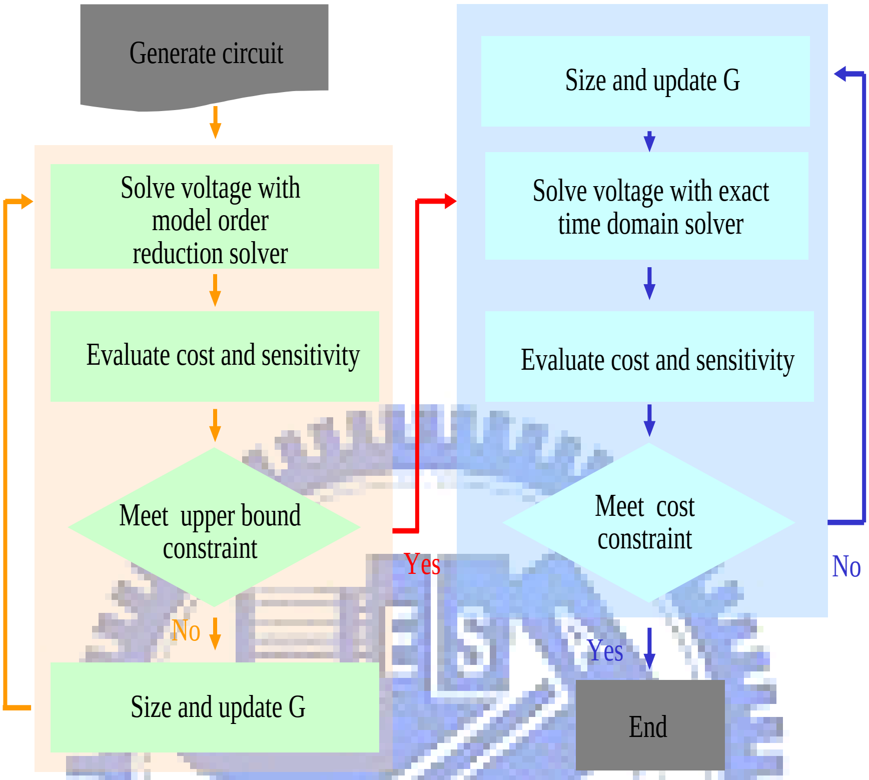

Generate circuit
Solve voltage with model order reduction solver
Evaluate cost and sensitivity
Meet upper bound constraint
No
Size and update G
Yes
Size and update G
Solve voltage with exact time domain solver
Evaluate cost and sensitivity
Meet cost constraint
No
Yes
End

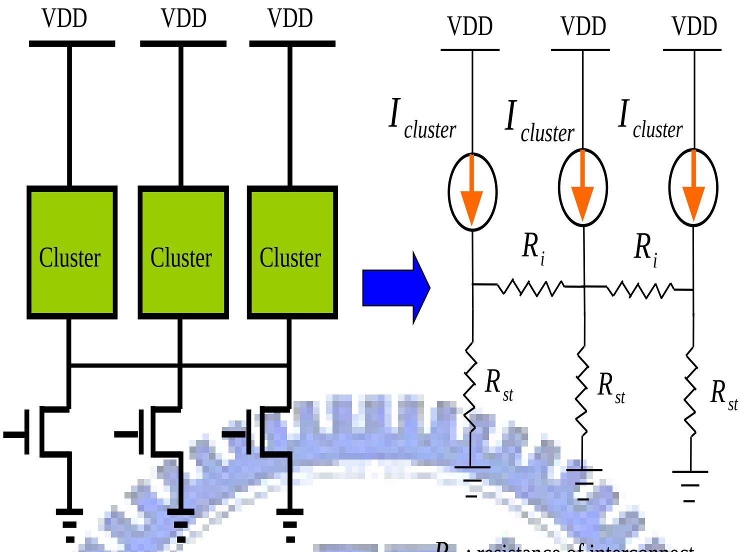
Fig. 3.2: The flowchart of the two-stage sizing algorithm.

- **Object:** Minimize the total width of sleep transistors subject to meet the maximum allowed threshold voltage constraint.

- **Output:** The width of each sleep transistor and the total width of all the sleep transistors.

3.2 DSTN Modeling

In the DSTN structure, the virtual ground line and sleep transistor can be modeled as equivalent resistances, and each cluster is modeled as a time-variant piece wise linear (PWL) waveform, so the DSTN structure can be transformed into an IR equivalent network as shown in Fig. 3.3. In the IR equivalent circuit, we employ Kirchhoff's Current



R_i : resistance of interconnect

R_{st} : resistance of sleep transistor

Fig. 3.3: The IR equivalent circuit of a DSTN structure

Law (KCL) to the node 2 as show in Fig. 3.4, where R_{sti} is the resistance of sleep transistor i, $V_i(t)$ is the voltage waveform at node i, R_{ij} is the resistance between node i and node j, and $I_i(t)$ is a current injected into node i. We then get an equation as

$$\begin{aligned} I_2(t) &= (V_2(t) - V_1(t))/R_{21} + (V_2(t) - V_3(t))/R_{23} + (V_2(t) - V_6(t))/R_{26} \\ &+ (V_2(t) - V_5(t))/R_{25} + V_2(t)/R_{st2}. \end{aligned} \tag{3.1}$$

Applying the KCL to each node of this equivalent IR circuit, the behavior of the system can be expressed by modified nodal analysis (MNA) formulation as

$$\begin{bmatrix} g_{11} & g_{12} & \cdots & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & \cdots & g_{nn} \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_2(t) \\ \vdots \\ V_n(t) \end{bmatrix} = \begin{bmatrix} I_1(t) \\ I_2(t) \\ \vdots \\ I_n(t) \end{bmatrix}, \tag{3.2}$$

where g_{ij}=$(-1/R_{ij})$ and g_{ii}=($\sum 1/R_{ij}$). Equation (3.2) can be reformulated as

$$\mathbf{G}(\mathbf{W^{st}})\mathbf{V}(\mathbf{t}, \mathbf{W^{st}}) = \mathbf{U}(\mathbf{t}), \tag{3.3}$$

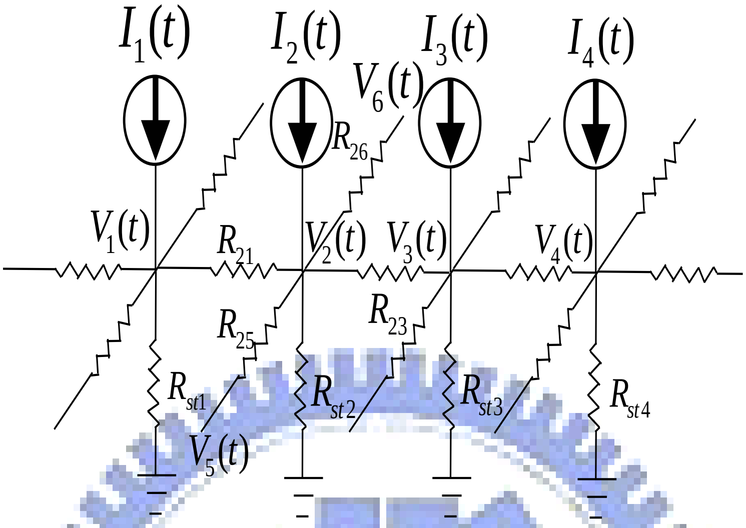


Fig. 3.4: DSTN equivalent circuit

where $\mathbf{G}(\mathbf{W}^{\mathbf{st}})$ is the conductance matrix, $\mathbf{U}(t)$ is the vector of current flowing through sleep transistors, $\mathbf{V}(t, \mathbf{W}^{\mathbf{st}})$ is the vector of voltage drop across sleep transistors and $\mathbf{W}^{\mathbf{st}}$ is the vector of width.

3.3 Definition of Cost

The integral of voltage waveform at node i above a maximum allowed threshold voltage V_i^{st*} is an efficient metric for the performance of each node in the DSTN structure. On the other hand, the cost at node i includes a time period. If we sum the cost of each node, it can be viewed as global view, the results of global view point is better than local view point. As shown in Fig. 3.5, the cost of node i can be defined as

$$\begin{aligned} c_i(\mathbf{W}^{st}) &\triangleq \int_0^T max\Big(V_i(t, \mathbf{W}^{st}) - V_i^{st*}, 0\Big)dt \\ &= \int_0^T q_i(t)(V_i(t, \mathbf{W}^{st}) - V_i^{st*})dt, \end{aligned} \tag{3.4}$$

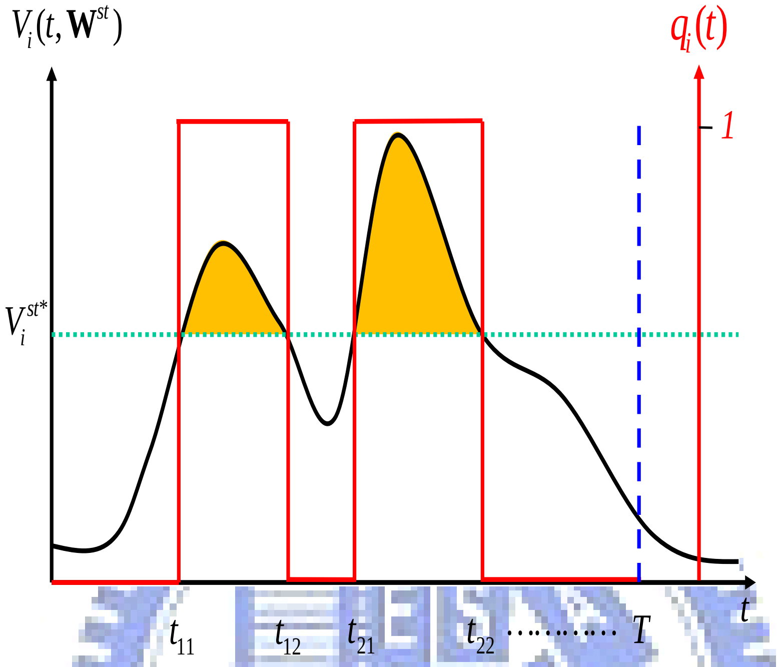


Fig. 3.5: The voltage waveform across a sleep transistor.

where $V_i(t, \mathbf{W}^{st})$ is voltage waveform at node i, and $\mathbf{W}^{st}$ is a vector of width. The $q_i(t)$ is a waveform which indicates the voltage violation intervals; for example, there are two unit pulses within time intervals $[t_{11}, t_{12}]$ and $[t_{21}, t_{22}]$ in $q_i(t)$ as shown in Fig. 3.5.

The cost of each node has been defined in Equation (3.4). Hence, the total cost of a DSTN design can be defined as

$$c(\mathbf{W}^{st}) \triangleq \sum_i c_i(\mathbf{W}^{st}). \tag{3.5}$$

Given a set of V_i^{st*}'s, each $V_i(t, \mathbf{W}^{st})$ in the DSTN design must be always less than V_i^{st*}, i.e. the cost of each node must be equal to zero. Hence, Equation (3.5) be equal to zero.

3.4 Definition and Computation of Integral Sensitivity

Integral sensitivity is utilized to size the sleep transistors, because the sensitivity indicates the effectiveness of sizing the sleep transistor. From global view point, we combine cost and sensitivity, and then use them to size sleep transistors. Finally, it can get better results.

In this work, the integral sensitivity of sleep transistor i is defined as S_i. It can be written as

$$S_i \triangleq \frac{\partial c}{\partial W_i^{st}} \tag{3.6}$$

where c is the total cost of DSTN.

To calculate each S_i, firstly, a new function $\tilde{c}(t, \mathbf{W}^{st})$ is defined as

$$\begin{aligned}\tilde{c}(t, \mathbf{W}^{st}) &= \sum_i \int_0^t q_i(\tau)(V_i(\tau, \mathbf{W}^{st}) - V_i^{st*})d\tau \\ &= \sum_i \int_0^t h_i(t-\tau)(V_i(\tau, \mathbf{W}^{st}) - V_i^{st*})d\tau,\end{aligned} \tag{3.7}$$

where $h_i(t-\tau) \triangleq q_i(\tau)$ for each i. It can be observed that $c(\mathbf{W}^{st}) = \tilde{c}(T, \mathbf{W}^{st})$. Equation (3.7) can be expressed as the follow vector form.

$$\tilde{c}(t, \mathbf{W}^{st}) = \int_0^t \mathbf{h}^T(t-\tau)(\mathbf{V}(t, \mathbf{W}^{st}) - \mathbf{V}^{st*})d\tau, \tag{3.8}$$

where $\mathbf{h}(t)$ is a functional vector with each entry i being $h_i(t)$, and $\mathbf{V}^{st*}$ is a vector with each entry equal to V_i^{st*}. By taking Laplace transform on both sides of Equations (3.8) and (3.3), we have

$$\mathbf{G}(\mathbf{W}^{st})\tilde{\mathbf{V}}(s, \mathbf{W}^{st}) = \tilde{\mathbf{U}}(s) \tag{3.9}$$

$$\hat{c}(s, \mathbf{W}^{st}) = \tilde{\mathbf{h}}^T(s)\left(\tilde{\mathbf{V}}(s, \mathbf{W}^{st}) - \frac{\mathbf{V}^{st*}}{s}\right), \tag{3.10}$$

where $\hat{c}(s, \mathbf{W}^{st})$, $\tilde{\mathbf{h}}(s)$, $\tilde{\mathbf{V}}(s)$ and $\tilde{\mathbf{U}}(s)$ are Laplace transforms of $\tilde{c}(t, \mathbf{W}^{st})$, $\mathbf{h}(t)$, $\mathbf{V}(t)$ and $\mathbf{U}(t)$, respectively. From Equation (3.9), we have $\tilde{\mathbf{V}} = \mathbf{G}^{-1}\tilde{\mathbf{U}}$, so Equation (3.10) can be formulated as

$$\hat{c}(s, \mathbf{W}^{st}) = \tilde{\mathbf{h}}^T(s)\left(\mathbf{G}^{-1}(\mathbf{W}^{st})\tilde{\mathbf{U}}(s) - \frac{\mathbf{V}^{st*}}{s}\right). \tag{3.11}$$

Therefore,

$$\begin{aligned}\frac{\partial \hat{c}}{\partial W_i^{st}} &= -\tilde{\mathbf{h}}^T \mathbf{G}^{-1} \frac{\partial \mathbf{G}}{\partial W_i^{st}} \tilde{\mathbf{V}} \\ &= \tilde{\mathbf{V}}^{aT} \frac{\partial \mathbf{G}}{\partial W_i^{st}} \tilde{\mathbf{V}},\end{aligned} \tag{3.12}$$

where $\tilde{\mathbf{V}}$ is the Laplace transform of the solution of original MNA equations, and $\tilde{\mathbf{V}}^a$ is the solution of adjoint MNA equations in the s-domain, i.e.

$$\mathbf{G}\tilde{\mathbf{V}} = \tilde{\mathbf{U}}, \tag{3.13}$$

$$\mathbf{G}^T\tilde{\mathbf{V}}^a = -\tilde{\mathbf{h}}. \tag{3.14}$$

Finally, the sensitivity of $c(\mathbf{W}^{st})$ with respect to an arbitrary parameter W_i^{st} is equal to

$$\begin{aligned}\frac{\partial c(\mathbf{W}^{st})}{\partial W_i^{st}} &= \left.\frac{\partial \tilde{c}(t, \mathbf{W}^{st})}{\partial W_i^{st}}\right|_{t=T} \\ &= \int_0^T \mathbf{V}^{aT}(t-\tau)\frac{\partial \mathbf{G}}{\partial W_i^{st}}\mathbf{V}(\tau)d\tau \\ &= \kappa \int_0^T V_i^a(T-\tau)V_i(\tau)d\tau \end{aligned} \tag{3.15}$$

In order to calculate each $\partial c(\mathbf{W}^{st})/\partial W_i^{st}$, we need to know the waveforms of $\mathbf{V}^a(t)$ and $\mathbf{V}(t)$ which can be done by applying the trapezoidal integral approximation to Equations (3.13) and (3.14) in the time domain, solving them easily with only two forward/backward substitutions at each time step. After solving the waveforms of $\mathbf{V}^a(t)$ and $\mathbf{V}(t)$ at each time step, we utilize Riemann's sum to estimate the sleep transistor sensitivity. In this thesis, we employ the model order reduction method to solve the matrix problem in the first stage because it is faster than the exact time domain solver. Details will be described in the next section.

For the reader's information, the $\partial c_i(\mathbf{W}^{st})/\partial W_i^{st}$ can be obtained by the above procedure, i is from one to the total number of nodes. Because the $\partial c_i(\mathbf{W}^{st})/\partial W_i^{st}$ is smaller than $\partial c(\mathbf{W}^{st})/\partial W_i^{st}$, if we employ the $\partial c_i(\mathbf{W}^{st})/\partial W_i^{st}$ to size sleep transistors i, it might result in oversizing, hence we utilize the $\partial c(\mathbf{W}^{st})/\partial W_i^{st}$ to size the sleep transistors, thereby obtaining better results.

3.5 Model Order Reduction Method

In this section we introduce model order reduction techniques which have proven to be very efficient for speeding up circuit analysis [7, 8]. IEKS (Improved Extended Krylov Subspace) method is one of these. IEKS was developed in [8], and it does not need to perform moment shifting for source waveform modeling. The major advantage of IEKS method is that its runtime is not proportional to the number of independent sources. Since the DSTN structure contains many current sources, our sensitivity computation is coupled with IEKS-based order reduction approach so that it can handle large scale DSTN design. In [7, 8], the authors analyzed circuits with piece wise linear (PWL) waveforms. In the present study we not only analyze circuits with PWL waveforms but also analyze circuits with several unit pulse waveforms. In section 3.5.2, we will describe how to calculate the moment with several unit pulse waveform.

3.5.1 Improved Extended Krylov Subspace Method(IEKS)

In Equation (3.3) we employ Laplace transformation on both sides and then obtain $\mathbf{G}\tilde{\mathbf{V}}(\mathbf{s}) = \tilde{\mathbf{U}}(\mathbf{s})$, where $\tilde{\mathbf{V}}(\mathbf{s})$ and $\tilde{\mathbf{U}}(\mathbf{s})$ are Laplace transform of $\mathbf{V}(\mathbf{t})$ and $\mathbf{U}(\mathbf{t})$. After calculating the moment, the orthonormal basis $\mathbf{X}$ of its extended Krylov subspace can be calculated. We then employ this basic to construct its order-reduced model by projecting the original system onto this subspace via congruent transformation. The reduction matrix will be calculated by $\hat{\mathbf{G}} = \mathbf{X}^{\mathbf{T}}\mathbf{G}\mathbf{X}$, $\hat{\mathbf{V}} = \mathbf{X}^{\mathbf{T}}\mathbf{V}$, $\hat{\mathbf{U}} = \mathbf{X}^{\mathbf{T}}\mathbf{U}$. The dimension of this new system is smaller than original system because the rank of $\hat{\mathbf{G}}$ is smaller than $\mathbf{G}$. Therefore the runtime is much less than the original circuit. Next we set up the system equations of reduced circuit and utilize the fast simulation method in [17] to get the waveform of $\hat{\mathbf{V}}(t)$. Then $\hat{\mathbf{V}}(t)$ is projected back to the original space to provide the approximate solution, $\mathbf{V}(\mathbf{t}) \approx \mathbf{X}\hat{\mathbf{V}}(\mathbf{t})$. Details of IEKS reduction procedure can be found in [6].

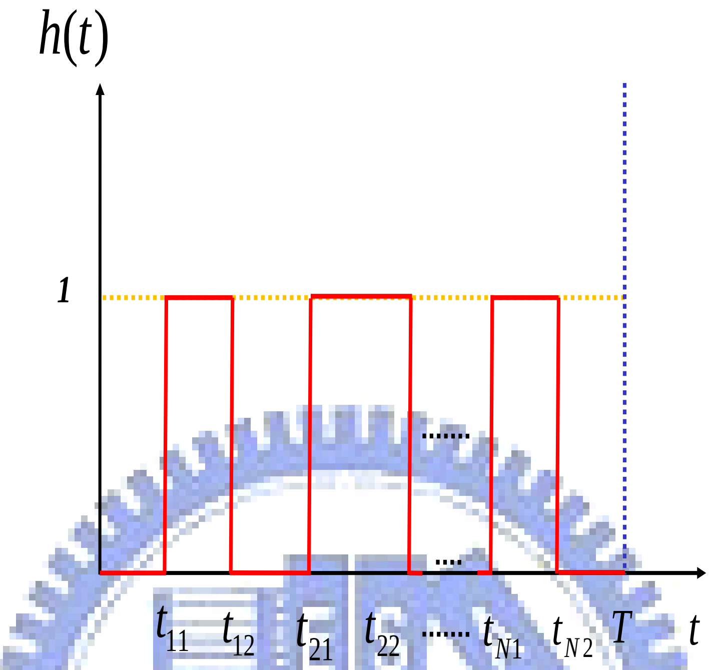


Fig. 3.6: A waveform contains several unit pulses

3.5.2 The Moment of Several Unit Pulses Waveform

In order to calculate the orthonormal basis $\mathbf{X}$, we need to firstly compute the moment form waveform. In [7, 8], the authors analyzed the circuit with piece wise linear (PWL) waveform. Now, we analyze the circuit with several unit pulses waveform in this session. Given a circuit with several unit pulses waveform, we can derivate the moment of the waveform. A unit pulse after shifting can be viewed as a unit step function after shifting subtracts another unit step function after shifting. If $h(t)$ is composed of N unit pulses as shown in Fig. 3.6, the details derivate as follows:

$$h(t) = \sum_{i=1}^{N} u(t - t_{i1}) - u(t - t_{i2}), \tag{3.16}$$

where $u(t)$ is a step function and $u(t-t_i)$ is a is a unit step function after shifting t_i. We take Laplace transformation on both sides and than have

$$\begin{aligned} L\{h(t)\} &= \sum_{i=1}^{N}\left(\frac{e^{-st_{i1}}}{s}-\frac{e^{-st_{i2}}}{s}\right) \\ &= \sum_{i=1}^{N}\frac{1}{s}\left\{\sum_{l=0}^{\infty}(-1)^l\cdot\left(\frac{t_{i1}^l}{l!}s^l\right)-\sum_{l=0}^{\infty}(-1)^l\cdot\left(\frac{t_{i2}^l}{l!}s^l\right)\right\} \\ &= \sum_{i=1}^{N}\frac{1}{s}\left\{\sum_{l=1}^{\infty}(-1)^l\cdot\left(\frac{t_{i1}^l-t_{i2}^l}{l!}s^l\right)\right\} \\ &= \sum_{i=1}^{N}\sum_{l=1}^{\infty}(-1)^l\cdot\left(\frac{t_{i1}^l-t_{i2}^l}{l!}s^{l-1}\right) \\ &= \sum_{l=1}^{\infty}(-1)^l\sum_{i=1}^{N}\left(\frac{t_{i1}^l-t_{i2}^l}{l!}s^{l-1}\right), \end{aligned} \tag{3.17}$$

whereupon we let $l=m+1$ and thus obtain

$$\sum_{l=1}^{\infty}(-1)^l\sum_{i=1}^{N}\left(\frac{t_{i1}^l-t_{i2}^l}{l!}s^{l-1}\right)=\sum_{m=0}^{\infty}(-1)^{m+1}\sum_{i=1}^{N}\left(\frac{t_{i1}^{m+1}-t_{i2}^{m+1}}{(m+1)!}s^m\right). \tag{3.18}$$

Therefore, the m-th moment of $h(t)$ can be calculated by

$$h_m=(-1)^{m+1}\sum_{i=1}^{N}\left(\frac{t_{i1}^{m+1}-t_{i2}^{m+1}}{(m+1)!}\right), \tag{3.19}$$

where m is the m-th moment. Fig. 3.7 shows the proposed moment calculation algorithm.

3.6 Two-Stage Sizing Algorithm

As shown in Fig. 3.4, the DSTN design optimization problem can be expressed as the following problem formulation.

"Given an IR circuit shown in Fig. 3.4 with a set of independent PWL current sources and the maximum allowed threshold voltage at each node, the problem is to simultaneously minimize the total width of sleep transistors and meet the maximum allowed threshold voltage constraints."

The flowchart of sizing algorithm is shown in Fig. 3.2, and its details are shown in Fig. 3.9. Our sizing algorithm has two stages. To speed up the optimization procedure,

Moment Calculation Algorithm

Input : A waveform $h(t)$ contains
N unit pulses $\{(t_{11}, t_{12}), (t_{21}, t_{22}) \cdots, (t_{N1}, t_{N2})\}$

Output : $\mathbf{h_m} = \{h_1, h_2, ..., h_m\}$, the first m moments of the source $h(t)$.

```
Begin
    For $i = 1 : m$
        $moment = 0$
        For $j = 1 : N$
            $moment+ = t_{j1}^{m+1} - t_{j2}^{m+1}$
        End For
        For $k = 1 : (m + 1)$
            $moment = \frac{moment}{k}$
        End For
        If $(m + 1)\ \%\ 2 = 0$
            $h_i = moment$
        Else
            $h_i = -moment$
    End For
End.
```

Fig. 3.7: Moment calculation algorithm for the pulse waveform.

in the first stage we use IEKS reduction method to approximate the solution of the equivalent DSTN circuit with many independent time-variant PWL current sources, and then calculate the solution of the adjoint system of this equivalent circuit with many independent pulse waveforms. After that, the cost of each node and the integral sensitivity of each sleep transistor are computed, as shown in sections 3.3 and sections 3.4. To potentially zero the cost of the selected sleep transistors and also minimize the increased width of a DSTN, W_i^{add} is computed as

$$W_i^{add} = \frac{c_i}{S_i}, \tag{3.20}$$

where c_i is the cost of node i, and S_i is the integral sensitivity of sleep transistor i.

By the above sizing method the new width of sleep transistor i becomes

$$W_i^{new} = W_i^{old} + W_i^{add}. \tag{3.21}$$

In the sizing procedure of transistor i, the influence of sizing sleep transistor i is greater than sizing the other sleep transistors. So if node i has maximum $V_i^{max\_vio}$, the best method to reduce the cost of node i is to size the sleep transistor i. In our method, we choose the sleep transistor with maximum $V_i^{max\_vio}$ to size, hence the sleep transistor oversizing issue is alleviated. However, the conductance matrix and its LU decomposition need to be recalculated after each sizing step. This recalculation procedure is time-consuming. To reduce the number of recalculation steps, several sleep transistors are simultaneously sized instead of only one sleep transistor. The details will be described in the next section. The above procedure is repeated until the maximum $V_i^{max\_vio}$ of the sleep transistors plus V_i^{st*} is less than the upper bound voltage V^{up}.

In the second stage, an accurate and efficient time domain solver [17] is used to calculate the cost function and its sensitivity. The procedure presented in the first stage is repeated until the total cost is equal to zero. Fig. 3.8 shown the voltage waveform of node i in the sizing procedure from beginning to end. Fig. 3.8 (a) and Fig. 3.8 (b) show the voltage variable at node i in the first stage. Fig. 3.8 (c) illustrates it meets upper bound voltage constraint, and then starting the second stage. Fig. 3.8 (d) shows the cost of node i is equal to zero.

The total width of sleep transistors and the execution time of the proposed two-stage sizing algorithm are dependent on the value of V^{up} in the first stage. From the experimental results shown that it can be observed that the execution time is reduced when we decrease V^{up}. This is because IEKS method is faster than the time domain solver [17]. On the other hand, the total width of sleep transistors is reduced when we choose a larger V^{up}. This is because the time domain solver [17] is more accurate than IEKS method.

3.7 Speed Up

In the sizing procedure, as runtime is increased with the circuit become large. In order to decrease the runtime of circuit analysis, we also proposed the speed up method for sizing sleep transistors.

In our algorithm, after solving the voltage of nodes at each time step, we then choose the sleep transistor i with the maximum $V_i^{max\_vio}$ and size its width. Finally we update conductance matrix G, then repeat above procedure until meet cost constraint or upper bound voltage constraint. When we solve voltage of nodes at each time step, we need to perform LU decomposition, this is time-consuming so we modify our algorithm. First, we calculate the $V_i^{max\_vio}$ of each sleep transistor. Then, these values are sorted into descending order. After that, the first N sleep transistors are chosen for sizing. It can be observed in the experimental results that increasing N can reduce the execution time, but the total width will increase a little.

3.8 Discussion

In the chapter, we have proposed the two-stage sizing algorithm by using cost and integral sensitivity. In the first stage, model order reduction mthod is empoly because it can speed up the analysis of large circuits. On the other hand, the upper bound voltage is employed in the first stage, it can decide the runtime of all sizing procedure and total width of sleep transistor. If the upper bound voltage is increased, the runtime of first stage is decrease, and the runtime of second stage is increased. This is because it quickly meets the upper bound constraint in the first stage, then break the first stage and starting the second stage. In the second stage, exact time domain is utilized, it can solve the exact solution and make sure that final solution is not violation the circuit constraint, but it is time-consuming. That is why increase upper bound voltage will increase total runtime. Comparing the performance of using two stage with using only the second stage, the runtime of the latter is more than the former, the experimental results will be seen in the next chapter, and it will show that our two-stage sizing algorithm can handle large circuits efficiently. In [4], a state-of-the-are method has been proposed, the authors just only consider the maximum $V_i^{max\_vio}$ at node i and then reduce the maximum $V_i^{max\_vio}$. In our method, on the other hand, we consider not only the cost of time period T but also consider the total cost of the nodes, i.e. we consider the global region. From the global view point, our method is outperform [4]. The experimental results will show our method is better than [4] in next chapter.

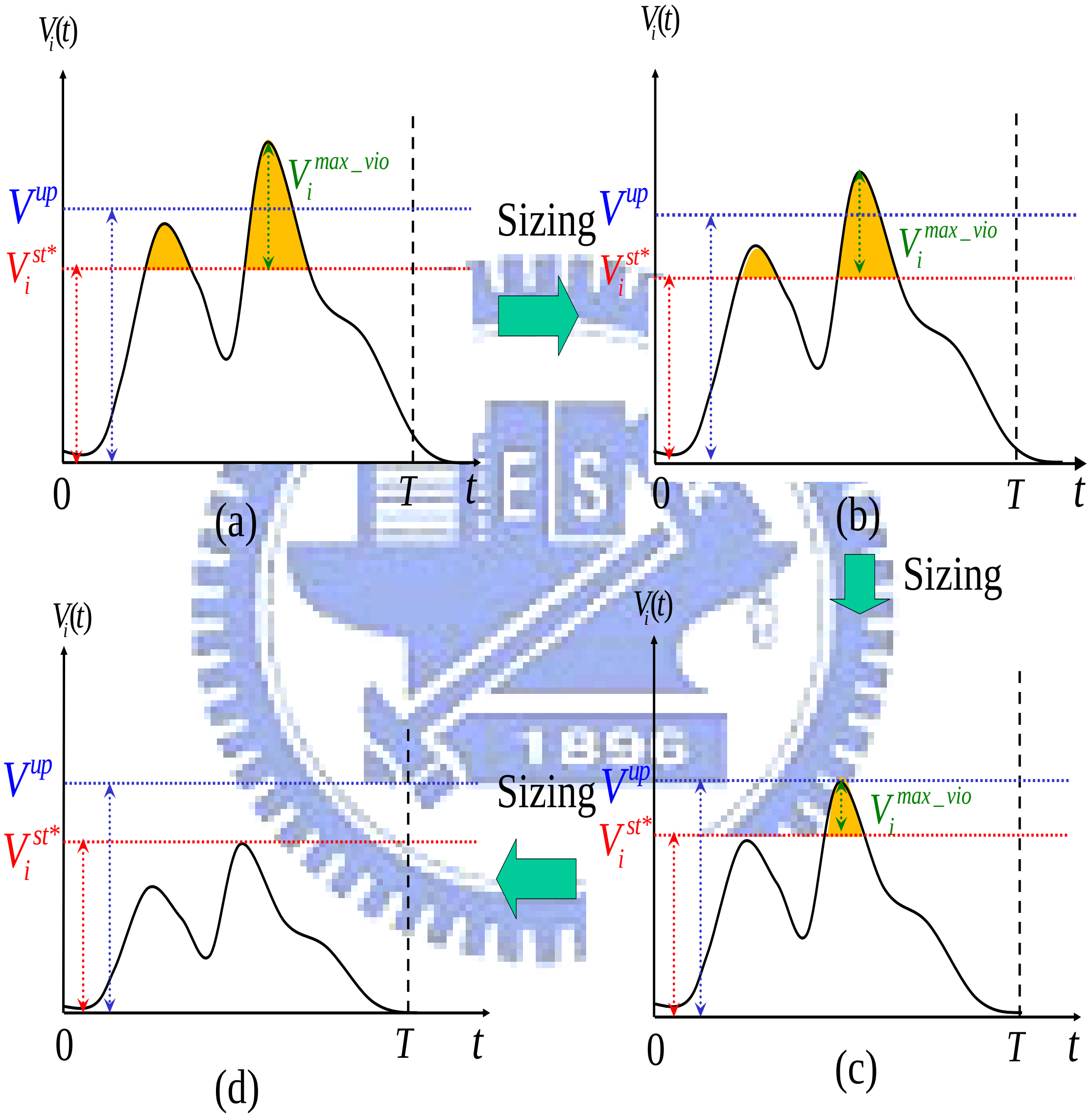


Fig. 3.8: (a)The voltage waveform of node i in the first sizing stage. (b) The $V_i^{max\_vio}$ of node i was reduced. (c)The voltage waveform of node i in the second stage. (d) The $V_i^{max\_vio}$ of node i is equal to zero.

Two-Stage Sizing Algorithm

Input : The numbers of sleep transistor K, the numbers of slected sleep transistors N, the upper bound voltage V^{up}, the maximum allowed threshold voltage V_i^{st*} for each node i, and minimum width of sleep transistors.

Output : Total width of sleep transistors.

```
Begin
    Construct conductance matrix $G$
    FIRST STAGE
    While
        Solve voltage at each time step with IEKS method
        Sort $V_i^{max\_vio}$ from maximum to minimum
        For $i = 1 : N$
            Estimate cost and sensitivity for node $i$
        End For
        If don't meet upper bound voltage constraint
            Select first N nodes in this order
            For $i = 1 : N$
                $W_i^{add} = \frac{c_i}{S_i} => W_i^{new} = W_i^{old} + W_i^{add}$
            End For
            Update conductance matrix $G$
        End If
        Else Break
    End While
    SECOND STAGE
    While
        Solve voltage at each time step with exact time domain solver
        Sort $V_i^{max\_vio}$ from maximum to minimum
        For $i = 1 : N$
            Estimate cost and sensitivity for node $i$
        End For
        If don't meet total cost constraint
            Select first N nodes in this order
            For $i = 1 : N$
                $W_i^{add} = \frac{c_i}{S_i} => W_i^{new} = W_i^{old} + W_i^{add}$
            End For
            Update conductance matrix $G$
        End If
        Else Break
    End While
End.
```

Fig. 3.9: The two-stage sizing algorithm.

Chapter 4

Experimental Results

The two-stage sizing algorithm is implemented in C++ language on a processor 3.2GHZ HP workstation with 32GB memory. The test circuits are randomly generated, and the numbers of nodes are from 400 to 5000. In the test circuit, the waveform of each current source contains several triangle waves with peak value randomly generated between 0.1mA and 5mA. Our V_{DD} is set to 1.3V, and the maximum allowed threshold voltage is set to 0.065V, which is 5% of the V_{DD}. The virtual ground resistances are randomly generated from 10 to 20Ω. Time period T is partitioned into 100 time steps.

The plots shown in Fig. 4.1 and Fig. 4.2 are the results for the test circuit with 2500 nodes. Fig. 4.1 illustrates the relation between the total width of sleep transistors, the upper bound voltage V^{up} and the number N of selected sizing sleep transistors. It can be observed that the total width decreases as V^{up} increases or N decreases. Fig. 4.2 illustrates the relation between the runtime, the upper bound voltage V^{up} and the number N of selected sizing sleep transistors. It can be observed that the runtime decreases as V^{up} decreases or N increases.

Circuit	Area (Width) μm			
	[4]		Two-Stage Sizing Algorithm	
no. of nodes	no. of time frames		no. of selected transistors (N)	
	TP	V-TP	1	4% of sleep transistors
400	558.0900	1919.5600	556.3900	556.7100
900	1162.4500	4565.2600	1158.6300	1159.4300
1225	1611.5000	6258.1000	1607.6300	1609.2400
1600	2045.8400	8952.9100	2038.0000	2043.2000
2025	2595.4100	11137.3000	2588.6300	2591.5400
2500	3191.0800	14893.6000	3179.9300	3183.6300
5000	6179.0300	32326.2000	6158.7500	6173.1900
Total	17343.4100	80052.9300	17287.9600	17316.9400

Table 4.1: Total width comparison for two-stage method and [4]

Circuit	The ratio of width			
	[4]		Two-Stage Sizing Algorithm	
no. of nodes	no. of time frames		no. of selected transistors (N)	
	TP	V-TP	1	4% of sleep transistors
400	1.0025	3.4481	0.9994	1.0000
900	1.0026	3.9375	0.9994	1.0000
1225	1.0014	3.8889	0.9990	1.0000
1600	1.0013	4.3818	0.9975	1.0000
2025	1.0015	4.2976	0.9989	1.0000
2500	1.0023	4.6782	0.9988	1.0000
5000	1.0009	5.2365	0.9977	1.0000
Avg.	1.0018	4.2669	0.9986	1.0000

Table 4.2: The ratio of width for two-stage method and [4].

In order to demonstrate that two-stage sizing algorithm is better than the state-of-the-art method [4], we also implement the method [4]. In Tables 4.1 and 4.2, the TP is the sizing method by using the uniform time frame partition proposed by [4] with 100 time frames, and the V-TP is the sizing method by using the variable time frame partition developed in [4] with 10 time frames. In our method, the V^{up} is set to be 0.2V in the first stage, and the number of selected sleep transistors is 4$\%$ of the total sleep transistors.

Table 4.1 and Table 4.2 demonstrates that the total width of the proposed sizing algorithm outperforms both the TP and V-TP methods. In fact, the total width of our two-stage sizing algorithm is slightly less than the TP for each test circuit, and the total width of V-TP averages over four times that of our proposed method. Table 4.3 and Table 4.4 show

Circuit	Runtime (s)			
	[4]		Two-Stage Sizing Algorithm	
no. of nodes	no. of time frames		no. of selected transistors (N)	
	TP	V-TP	1	4% of sleep transistors
400	7.4900	3.0400	102.7400	6.9500
900	42.2700	17.7900	665.3200	18.6400
1225	83.9400	34.9500	1279.7700	26.1600
1600	173.8900	80.5900	2601.6200	44.4700
2025	444.7400	173.3200	6353.0800	71.8400
2500	1068.8400	269.9600	14391.0000	101.2900
5000	9876.9500	2764.0900	153513.0000	749.2800
Total	11698.1200	3343.7400	178906.5000	1018.6300

Table 4.3: Runtime comparison for two-stage method and [4].

Circuit	The ratio of runtime			
	[4]		Two-Stage Sizing Algorithm	
no. of nodes	no. of time frames		no. of selected transistors (N)	
	TP	V-TP	1	4% of sleep transistors
400	1.0777	0.4374	14.7827	1.0000
900	2.2677	0.9544	35.6931	1.0000
1225	3.2087	1.3360	48.9209	1.0000
1600	3.9103	1.8122	58.5028	1.0000
2025	6.1907	2.4126	88.4337	1.0000
2500	10.5523	2.6652	142.0772	1.0000
5000	13.1819	3.6889	204.8807	1.0000
Avg.	5.7699	1.9909	84.7558	1.0000

Table 4.4: The ratio of runtime for two-stage method and [4].

that two-stage sizing algorithm is faster than both the TP and V-TP methods, specifically averaging 5.77× faster than the TP and 1.99× faster than the V-TP.

In Table 4.4, comparing TP or V-TP with two-stage method, it can be observed as the circuit become large, the ratio of runtime increases. We moreover compare, using our method, sizing based on selecting only one sleep transistor relative to resizing based on selecting 4% of the total sleep transistors. It is found that the total width of the former is less than the latter, but the run time of the latter is 84.75× faster than the former. It is thus shown that selecting multiple sleep transistors for concurrent sizing is a very fast and reasonably effective methodology.

Table 4.5 shows that the total width obtained using only the second stage is less than

Circuit	Area (Width) μm			
	Only execution in second stage		Two-Stage Sizing Algorithm	
no. of nodes	no. of selected transistors (N)		no. of selected transistors (N)	
	1	4% of sleep transistors	1	4% of sleep transistors
400	556.0700	556.1500	556.3900	556.7100
900	1157.6400	1157.9000	1158.6300	1159.4300
1225	1605.8300	1606.1600	1607.6300	1609.2400
1600	2036.4800	2037.2300	2038.0000	2043.2000
2025	2586.2000	2586.6700	2588.6300	2591.5400
2500	3177.5600	3178.3000	3179.9300	3183.6300
Total	11119.7800	11122.4100	11129.2100	11143.7500

Table 4.5: Total width comparison for two-stage method and only execution in second stage.

Circuit	The ratio of width			
	Only execution in second stage		Two-Stage Sizing Algorithm	
no. of nodes	no. of selected transistors (N)		no. of selected transistors (N)	
	1	4% of sleep transistors	1	4% of sleep transistors
400	0.9988	0.9989	0.9994	1.0000
900	0.9984	0.9986	0.9993	1.0000
1225	0.9979	0.9981	0.9990	1.0000
1600	0.9967	0.9971	0.9975	1.0000
2025	0.9979	0.9981	0.9988	1.0000
2500	0.9981	0.9983	0.9988	1.0000
Avg.	0.9980	0.9982	0.9988	1.0000

Table 4.6: The ratio of width for two-stage method and only execution in second stage.

when using the full two stage sizing method. This is because the exact time domain solver provides a best candidate for choosing and sizing sleep transistors. Hence, the total width can be reduced. Table 4.7 shows that the two stage sizing method is faster than using only the second stage. It also reveals that model order reduction is an effective method for circuit analysis. In Table 4.8, it can be observe that comparing only execution in second stage with two-stage method, as the circuit become large, the ratio of runtime increases. Finally, memory usage of our method is compared with [4], showing that the memory usage of our method is more than [4], with the results shown in Table 4.9.

Circuit	Runtime (s)			
	Only execution in second stage		Two-Stage Sizing Algorithm	
no. of nodes	no. of selected transistors (N)		no. of selected transistors (N)	
	1	4% of sleep transistors	1	4% of sleep transistors
400	155.3800	8.7600	102.7400	6.9500
900	989.7800	27.9500	665.3200	18.6400
1225	1884.7800	39.5800	1279.7700	26.1600
1600	3512.2200	70.3600	2601.6200	44.4700
2025	7579.9400	150.9600	6353.0800	71.8400
2500	20668.500	213.8300	14391.0000	101.2900
Total	34790.6000	511.4400	25393.5300	269.3500

Table 4.7: Runtime comparison for two-stage method and only execution in second stage.

Circuit	The ratio of runtime			
	Only execution in second stage		Two-Stage Sizing Algorithm	
no. of nodes	no. of selected transistors (N)		no. of selected transistors (N)	
	1	4% of sleep transistors	1	4% of sleep transistors
400	22.3568	1.2604	14.7827	1.0000
900	53.0998	1.4995	35.6931	1.0000
1225	72.0481	1.5129	48.9208	1.0000
1600	78.9795	1.5821	58.5028	1.0000
2025	105.5114	2.1013	88.4337	1.0000
2500	204.0527	2.1111	142.0772	1.0000
Avg.	89.3414	1.6779	64.73508	1.0000

Table 4.8: The ratio of runtime for two-stage method and only execution in second stage.

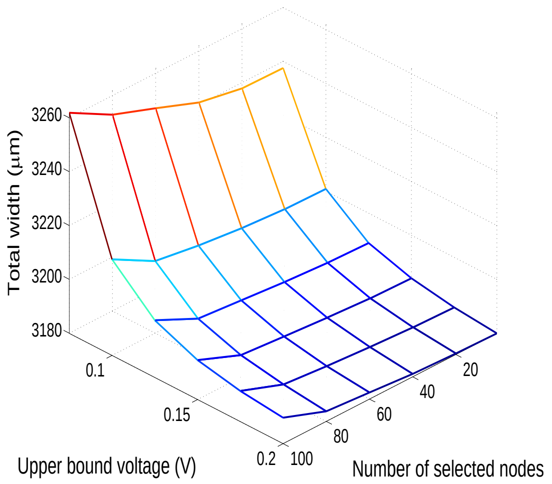


Fig. 4.1: Total width relation between the number of selected sleep transistors and the upper bound voltage.

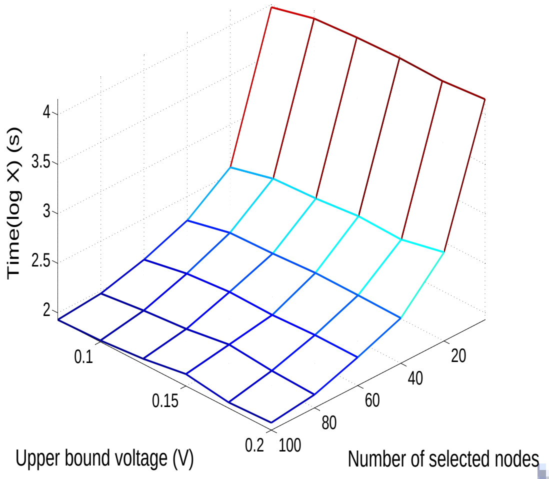


Fig. 4.2: Runtime relation between the number of selected sleep transistors and the upper bound voltage. The X is the execution time.

Circuit	Memory (MB)			
	[4]		Two-Stage Sizing Algorithm	
no. of nodes	no. of time frames		no. of selected transistors (N)	
	TP	V-TP	1	4% of sleep transistors
400	1.18	2.68	3.82	3.98
900	4.62	3.92	6.45	6.54
1225	5.66	4.70	8.09	8.21
1600	7.06	5.81	10.51	10.06
2025	8.40	6.82	12.61	11.90
2500	10.09	8.12	14.56	14.76
5000	19.00	8.89	26.03	27.71

Table 4.9: Memory usage comparison.

Chapter 5

Conclusion

In this thesis, we have presented a novel approach to size sleep transistors. In order to speed up circuit analysis, we employ model order reduction to solve voltage drop of sleep transistors, and select more than one sleep transistor to size in a loop, this can effective reduce execution time. On the other hand, we combine the cost and integral sensitivity and utilize them to size sleep transistor. In global view point,the answer of it is better than local view point, so the experimental results show that our total width of sleep transistors is less than [4], and execution time is faster than [4].

Bibliography

[1] F. Fallah, and M. Pedram, ”Standby and active leakage current control and minimization in CMOS VLSI circuits,” IEICE Trans. on Electronics, vol. E88-C, pp. 509-519, April 2005.

[2] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, ”Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” Proc. of the IEEE vol. 91, no. 2, pp. 305 - 327, February 2003.

[3] V. De, Y. Ye, A. Keshavarzi, S. Narendra, J. Kao, D. Somasekhar, R. Nair, and S. Borkar, ”Techniques for leakage power reduction,” in Design of High-Performance Microprocessor Circuits, A. Chandrakasan, W. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE, 2001, ch. 3, pp. 52-55.

[4] D. S. Chiou, D. C. Juan, Y. T. Chen, and S. C. Chang, ”Fine-grained sleep transistor sizing algorithm for leakage power minimization,” Proc. of ACM/IEEE Design Automation Conference (DAC-2007), 2007, pp. 81-86,.

[5] http://asic-soc.blogspot.com/2008/03/leakage-power-trends.html

[6] H. Chang, and S. S. Sapatnekar, ”Full-chip analysis of leakage power under process variations, including spatial correlations,” Proc. of ACM/IEEE Design Automation Conference (DAC-2005), 2005, pp. 523-528.

[7] Y. M. Lee, Y. Cao, T. H. Chen, J. Wang, and Charlie C. P. Chen, ”HiPRIME: Hierarchical and passivity preserved interconnect macromodeling engine for RLKC power

delivery,” IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems (TCAD), vol. 24, no. 6, pp. 797-806, June, 2005.

[8] Y. Cao, Y. M. Lee, T. H. Chen, and Charlie C. P. Chen, ”HiPRIME: Hierarchical and passivity reserved interconnect macromodeling engine for RLKC power delivery,” Proc. of ACM/IEEE Design Automation Conference (DAC-2002), 2002, pp. 379-384.

[9] C. Long, and L. He, ”Distributed sleep transistor network for power reduction,” IEEE Transaction on VLSI systems, vol. 12, No. 9, pp 181-186, September 2004.

[10] 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 J. Solid-State Circuits, vol. 31, issue 11, pp. 1795-1802, November 1996.

[11] J. Kao, S. Narendra, and A. Chandrakasan, ”MTCMOS hierarchical sizing based on mutual exclusive discharging patterns,” Proc. of ACM/IEEE Design Automation Conference (DAC-1998),1998, pp. 495-500.

[12] 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 ACM/IEEE Design Automation Conference (DAC-2002), 2002, pp. 480-485.

[13] J. Kao, A. Chandrakasan, and D. Antoniadis, ”Transistor sizing issues and tool for multi-threshold CMOS technology,” Proc. of ACM/IEEE Design Automation Conference (DAC-1997) , 1997, pp. 409-414.

[14] D. S. Chiou, S. H. Chen, S. C. Chang, and C. Yeh, ”Timing driven power gating,” Proc. of ACM/IEEE Design Automation Conference (DAC-2006), 2006, pp. 121-124.

[15] 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 (TCAD), Vol. 22, issue 10, pp. 1324-1342, June 2003.

[16] A. Ramalingam, B. Zhang, A. Devgan, and D. Z. Pan, ”Sleep transistor sizing using timing criticality and temporal currents,” Proc. of ACM/IEEE Asia and South Pacific Design Automation Conferernce (ASP-DAC-2005), vol. 2, pp. 1094-1097, Jan. 2005.

[17] T. H. Chen, and C. C. Chen, ”Efficient large-scale power grid analysis based on preconditioned Krylov subspace iterative methods,” Proc. Design Automation Conference (DAC-2001), 2001, pp. 559-562.