An Efficient Wake-up Schedule during Power Mode Transition 
Considering Spurious Glitches Phenomenon

Yu-Ting Chen, Da-Cheng Juan, Ming-Chao Lee, Shih-Chieh Chang
Department of CS, National Tsing Hua University, Hsinchu, Taiwan
chao@nthucad.cs.nthu.edu.tw, scchang@cs.nthu.edu.tw

ABSTRACT

Abstract—During the power mode transition, a large surge current may lead to the malfunctions in a power-gating design. In this paper, we introduce several important properties of the surge current during the power mode transition for the Distributed Sleep Transistor Network (DSTN) designs. Based on these properties, we propose an accurate estimation of surge current and provide an efficient schedule on the DSTN structure. Our experiment achieved significantly better results than previous works—on average, 332 times wake-up time reduction and 35.48% less energy loss during the power mode transition.

I. INTRODUCTION

In deep sub-micron CMOS technology, leakage increases exponentially and becomes a significant drain on total power consumption. Among the leakage reduction techniques, the power-gating technique has become one of the most effective methods. Recently, many industrial power-gating designs have adopted the Distributed Sleep Transistor Network (DSTN) structure [4][6].

During the sleep period, all internal devices and the virtual ground are gradually charged to $V_{DD}$ because of the leakage in the low $V_{th}$ devices. Therefore, when sleep transistors are turned on, a sudden discharge of the accumulated charges of internal devices leads to a large current, called a surge current, flowing through the sleep transistors to ground. The excessive surge current causes the Ldi/dt, IR drops and electromigration, which greatly affect the reliability and performance of a circuit [5]. Since the current flowing through sleep transistors is proportional to the total size of the turned-on sleep transistors, the number and the timing to turn on sleep transistors should be restricted to avoid the excessive surge current. Hence, the turn-on sequence of the sleep transistors, called the wake-up schedule, has become a major challenge to constrain the surge current in a power gating design. In [3], researchers proposed a method to turn on one sleep transistor per clock cycle, from the smallest size to the largest size.

We now describe two important observations during the power mode transition. We illustrate the first observation with Figure 1. Let us consider the waveforms of the surge currents under two different wake-up schedules. Before the 4000th nanosecond, the circuit is in the sleep mode. The dotted line stands for the surge current under an All-On schedule that turns on all transistors simultaneously at the 4000th nanosecond. The bold line presents the surge current under a One-by-One schedule that turns on sleep transistors one by one in a one-nanosecond time interval. Here, a dilemma scenario arises. On the one hand, the maximum surge current of the All-On schedule is 53.2 times larger than that of the One-by-One schedule. On the other hand, the wake-up time of the All-On schedule is 57.1% shorter than that of the One-by-One schedule. Conventionally, this trade-off between the wake-up time and the surge current has been modeled as a scheduling problem under a designer-specified surge-current constraint.

The other important observation is that spurious glitches are the major factors in total energy loss during the power mode transition. Figure 2 shows the voltage waveforms of an industrial design. The bold line stands for the voltage of an internal node $A (V_A)$ whereas the dotted line presents the voltage of $V_{GND} (V_{GND})$. Initially, node $A$ and $V_{GND}$ are charged close to $V_{DD}$ in the sleep mode. During the power mode transition, $V_{GND}$ needs sufficient time to stabilize to ground in practice. Meanwhile,
several spurious glitches occur before node $A$ reaches its final value. In this case, it takes 250ns for $V_{GND}$ to become stable. Many spurious glitches can occur during this period. According to our experiments, on average, 75.71% of the total discharging energy comes from spurious glitches that greatly increase the wake-up time.

In this paper, we analyze the behaviors of the surge current in DSTN designs. Based on the analyses, we devise a wake-up schedule that significantly reduces the wake-up time. To deal with spurious glitches, we also propose an intelligent technique that reduces the total energy loss and further improves the wake-up time. In comparison to [3], on average, our method achieves 332 times reduction in the wake-up time and reduces energy loss from spurious glitches by 35.48%.

II. PRELIMINARIES

![Figure 3. Current distribution of the DSTN design.](image)

In this section, we introduce the features of the DSTN structure and the calculation of surge current. Figure 3 shows a DSTN design with three logic clusters. Each cluster is connected to the corresponding sleep transistor $ST_i$ and to other sleep transistors by the virtual ground ($V_{GND}$). $V(ST_i)$ stands for the node voltage, which is connected to the corresponding $ST_i$ on $V_{GND}$. We would like to point out that $V(ST_i)$ may vary from one to another due to the current discharging balance phenomenon [4]. In [2], the researchers estimate the current flowing through turned-on $ST_i$, called $I_{turnon}(ST_i)$, in the active mode, but not during the mode transition.

In a DSTN design, a surge current that occurs during the power mode transition can be calculated as the summation of the current flowing through each turned-on sleep transistor, which can be expressed as Eq(1):

$$\text{surge current} = \sum I_{turnon}(ST_i) \quad \text{for all } i$$  \hspace{1cm} \text{Eq(1)}

III. SURGE CURRENT ANALYSIS

A. SURGE CURRENT VERSUS WAKE-UP TIME

According to [1][3], the wake-up time is formally defined as:

$$\text{wake-up time} = \max[T_{stable}(ST_i), T_{turnon}(ST_i)] \quad \forall i \implies T_{over} \quad \text{Eq(2)}$$

where $T_{stable}(ST_i)$ is the time when $V(ST_i)$ is stable within ±5% of nominal; $T_{turnon}(ST_i)$ stands for the time when $ST_i$ is turned on. Note that the wake-up time begins to be counted after any sleep transistor is turned on ($T_{begin}$).

Now we discuss how a surge current influences the wake-up time. The magnitude of a surge current determines the speed of discharging the accumulated energy. Hence, a short wake-up time can be achieved with a large surge current. An efficient wake-up schedule aims to drive the surge current to approach but not to exceed the designer-specified surge current constraint.

B. PROBLEM FORMULATION

Our problem formulation is shown as follows. First, the following three inputs are given: (1) a set of practical sizes for $ST_i$, $W(ST_i)$ [2][3][4], (2) the surge-current constraint ($SC\_CONSTRAINT$), and (3) a wake-up vector [5]. The initial condition is that all $ST_i$s are turned off. The decision variable is $T_{turnon}(ST_i)$. Again, $T_{turnon}(ST_i)$ stands for the time when $ST_i$ is turned on. The objective function is to minimize the wake-up time defined in Eq(2). Finally, the surge current at all times must satisfy $SC\_CONSTRAINT$.

C. SURGE CURRENT VERSUS VIRTUAL GROUND

In this section, we present the key factors affecting the estimation of a surge current. According to Eq(1), the surge current is the summation of $I_{turnon}(ST_i)$ at a given time. Hence, let us focus on the calculation of $I_{turnon}(ST_i)$ during the saturation region and the linear region. The equations are shown as Eq(3) and Eq(4) respectively.

$$I_{turnon}(ST_i) = k_c \frac{W(ST_i)}{L} \left[ V_{DD} - V_{TH} \right] (1 + \lambda V(ST_i)) \quad \text{Eq(3)}$$

$$I_{turnon}(ST_i) = k_c \frac{W(ST_i)}{L} \left[ V_{DD} - V_{TH} \right] W(ST_i) \left( 1 - \frac{V(ST_i)}{2} \right) \quad \text{Eq(4)}$$

where $W(ST_i)$ is the width of $ST_i$, $k_c$ is the process transconductance, $L$ is the channel length, $V_{TH}$ is the threshold voltage, $\lambda$ is the channel-length modulation parameter. Under a given set of $W(ST_i)$, the magnitude of $I_{turnon}(ST_i)$ depends on $V(ST_i)$ which equals the potential difference across $ST_i$.

We now describe a key characteristic of $V(ST_i)$. Empirically, $V(ST_i)$ is strictly decreasing in DSTN designs. We have simulated a large quantity of benchmarks under different wake-up schedules, finding that the empirical property holds in all DSTN designs.

IV. WAKE-UP SCHEDULES FOR WAKE-UP TIME MINIMIZATION CONSIDERING SPURIOUS GLUTCHES

A. AN EFFICIENT WAKE-UP SCHEDULE FOR WAKE-UP TIME MINIMIZATION

In Figure 4, since $V(ST_i)$ varies over time, we need to expand $V(ST_i)$ into $V(ST_i, t=t_0)$, which stands for the value of $V(ST_i)$ at $t = t_0$. Similarly, we expand $I_{turnon}(ST_i)$ into $I_{turnon}(ST_i, t=t_0)$. We assume that the surge-current constraint, $SC\_CONSTRAINT$, is set to 100mA and $W(ST_i)$s are given. Initially, before $t = 30$, both $ST_i$
and $ST_2$ are turned off. Again, we assume that $V(ST_1, t=30)$ and $V(ST_2, t=30)$ have been charged to $V_{DD}$. Let the wake-up process begin at $t = 30$, i.e. $T_{begin} = 30$. According to Eq(3)(4), we can calculate both $I_{turon}(ST_1, t=30)$ and $I_{turon}(ST_2, t=30)$. Then we can determine which STs can be turned on while still satisfying $SC\_CONSTRAINT$. In Figure 4(a), we have $I_{turon}(ST_1, t=30) = 90\text{mA}$ and $I_{turon}(ST_2, t=30) = 60\text{mA}$. To satisfy $SC\_CONSTRAINT$, we can turn on either $ST_1$ or $ST_2$ but not both, at $t = 30$. Let us choose to turn on $ST_1$ because $I_{turon}(ST_1, t=30)$ is larger than $I_{turon}(ST_2, t=30)$ and may lead to a shorter wake-up time. Therefore, we have $T_{turon}(ST_1) = 30$. Moreover, since $V(ST_1)$ is strictly decreasing, $V(ST_1, t=30)$ is the largest value of $V(ST_1, t \geq 30)$. We stress that this strictly decreasing characteristic of $V(ST)$ on the DSTN structure is critical because it can be applied to estimate the upper bound of a surge current. As a result, $I_{turon}(ST_1, t=30)$ is also the largest value of $I_{turon}(ST_1, t \geq 30)$ according to Eq(3)(4). Hence, $SC\_CONSTRAINT$ remains satisfied before we turn on the next STs. In our technique, if any ST still remains turned off, we will iteratively check and might turn on several STs after a certain time interval. The time interval is empirically decided. In our experiment, we set the time interval to 30ps. Hence, the next time interval is between $t = 30$ and $t = 60$. Because $ST_1$ remains turned off, we will simulate the design with $ST_1$ turned on between $t = 30$ and $t = 60$. We use simulations to obtain $V(ST_1)$. Figure 4(b) shows the second iteration of our technique. After the SPICE-like simulator updates the waveforms of $V(ST_1)$, we identify that $I_{turon}(ST_1, t=60) + I_{turon}(ST_2, t=60) = 90\text{mA}$, which will not exceed the $SC\_CONSTRAINT$. Therefore, we can turn on $ST_2$ at $t=60$ and have $T_{turon}(ST_2) = 60$. However, $V(ST_1, t=60)$ and $V(ST_2, t=60)$ are not yet stable in this time interval. At the end of the second iteration, we continue the simulation with both $ST_1$ and $ST_2$ turned on. Figure 4(c) shows that $V(ST_1, t=90)$ and $V(ST_2, t=90)$ are stable within ±5% of nominal. Therefore, $T_{stable}(ST_1)$ and $T_{stable}(ST_2)$, defined in Section 3.1, are set to 90. As a result, since $T_{begin} = 30$, the wake-up time is 90ps – 30ps = 60ps according to Eq(2).

Figure 5 presents the details of our schedule algorithm for Wake-up Time Minimization (WTM). From step 10 to step 14 of Figure 5, we aim to turn on several additional STs to have surge current as large as possible without exceeding $SC\_CONSTRAINT$. Note that this problem can be transformed into a well-known knapsack problem, which can be solved efficiently by the dynamic programming technique.

**Algorithm:** Wake-up Schedule ($W(ST_i)$, $SC\_CONSTRAINT$)

1: Output: A set of decision variables $T_{turon}(ST_i)$
2: for $i$ ← 1 to NUM_ST do
   /* step 1: initialization */
3:      $ST_i$ ← OFF;
4: end for
5: $t ← T_{begin}$; $surge\_current ← 0$;
6: repeat
   /* step 2: scheduling */
7:      update $V(ST, t)$ for all $i$ according to simulation results;
8:      update $I_{turon}(ST_i, t)$ for all $i$ according to Eq(3)(4);
9:      Apply the dynamic programming technique on OFF STs
10: For each OFF ST, which can be turned on do
11:     $ST_i$ ← ON;
12:     $T_{turon}(ST_i) ← t$;
13:     update $surge\_current ← \sum I_{turon}(ST_i, t)$ for all ON STs;
14: end for
   /* step 3: scheduling */
15: do simulation; $t ← t + time\_interval$;
16: until $ST_i$ is ON for all $i$
17: return $T_{turon}(ST_i)$ for all $i$;

Figure 6. A turned-on sequence considering the physical placement of sleep transistors.

Before taking spurious glitches into consideration, we focus on physical implementation of sleep transistors [6], which are normally placed in order. In our model, sleep transistors are placed at the ends of a row. Therefore, if there are five rows, there will be five sleep transistors aligned in order as shown in Figure 6. Typically, a sleep/wake-up signal is provided from a power...
management unit. Turning on sleep transistors without conforming to their order of physical placement may lead to a large routing area due to complicated power management units. As a result, in our wake-up schedule, we restrict the turn-on sequence of sleep transistors as follows. First, in our formulation, sleep transistors are numbered from $ST_1$ to $ST_k$ where $a < b < k$. After that, only consecutive $ST$s whose positions are next to $ST_1$ or $ST_k$ can be turned on. Figure 6 shows a turn-on sequence ($\{ST_3, ST_5, ST_7, ST_9\}$) following the above formulation. In the example, $ST_7$ is turned on first followed by $ST_5$ and $ST_9$ together. Finally, $ST_2$ and $ST_4$ are turned on at the same time.

We observe that a sleep transistor should have higher turn-on priority if the sleep transistor is connected to more logic gates close to primary inputs or the sleep transistor is close to the middle position in the layout. Here, we provide a simple yet effective heuristic, Intelligent Wake-up Time Minimization (IWMT). Let us explain the differences between IWMT and WTM in Section 4.1. The new algorithm IWTM calculates $\text{composit}_\text{weight}(ST_i)$ as the turn-on priority for the sleep transistors. The cost function, $\text{composit}_\text{weight}(ST_i)$ is calculated as $\text{composit}_\text{weight}(ST_i) = \text{topological}_\text{weight}(ST_i) + \text{position}_\text{weight}(ST_i) + \text{width}_\text{weight}(ST_i)$, where $\text{topological}_\text{weight}(ST_i) = \sum \left\{ \left( \text{level of gate } j \right) \right\}$, $\text{position}_\text{weight}(ST_i) = \text{middle_pos} - \text{pos}(ST_i) - \text{middle_pos}$ and $\text{width}_\text{weight}(ST_i) = \text{width}(ST_i) / \text{width}_\text{avg}$ represent three major parts which impact the wake-up time and the energy loss of a schedule. Then, we aim to turn on a high priority $ST_i$ if the summation of $I_{\text{turnon}}(ST_i, t)$ and $\text{surge}_\text{current}$ is smaller than or equal to SC\_CONSTRAINT. After that, we begin our iterative technique to accomplish the turn-on schedule. We can turn on prioritized $ST$s only when those prioritized $ST$s have turned-on neighbors and satisfy SC\_CONSTRAINT at the same time. The rest parts of IWTM are the same as WTM in Figure 5.

V. EXPERIMENTAL RESULTS

We re-implemented the method of [3] and compared that with our WTM and IWTM on the DSTN structure. In our experiments, we use the TSMC 90nm CMOS technology process. The time interval is set to 30ps for simulation. Additionally, we use the maximum instantaneous current (MIC) as the surge current constraint. We also insert decoupling capacitances in the experiments.

Table 1 shows our experimental results. The 3rd, 4th and 5th column shows the wakeup time and the 6th, 7th and 8th column shows the energy consumption during the power mode transition among [3], WTM, and IWTM. The surge current constraints are met in all cases. On average, the wake-up time of our IWTM is 332 times faster than that of [3]. Also, IWTM has 35.48% less energy loss than [3]. It is worth mentioning that the runtime of our flow can finish within ten minutes on all ISCAS benchmarks and within five hours on the AES (Advanced Encryption Standard) design.

VI. CONCLUSIONS

We have presented an effective and practical wake-up schedule, IWMT, on the DSTN structure. The main idea is to apply the strictly decreasing property of the virtual ground on the surge current estimation. We also minimize the energy loss from spurious glitches by IWMT considering physical implementation issues. The results show that our IWMT schedule, compared with [3], can achieve 332 times greater wake-up time reduction and reduce energy loss by 35.48% more.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Economic Affairs of Taiwan (96-EC-17-A-01-S1-038).

REFERENCES


