

# BTI-Gater: An Aging-Resilient Clock Gating Methodology

Liangzhen Lai

Vikas Chandra

Robert Aitken

Puneet Gupta

**Abstract**—Negative- and Positive-Bias Temperature Instability (N/PBTI) have become one of the most important reliability issues in modern semiconductor technology. N/PBTI-induced degradation depends heavily on workload, which causes imbalanced degradation and additional clock skew for clock distribution networks with clock gating features. In this work, we first analyze the effects of N/PBTI on clock paths with different clock gating use cases. Then cross-layer solutions are proposed to reduce N/PBTI-induced clock skew. Two Integrated Clock Gating (ICG) cell circuits are proposed to alternate clock idle state between logic high and logic low for each clock gating operation. A skew mitigation methodology is also proposed to select the appropriate ICG cells based on the architecture and microarchitecture context. An example of sleep scheduling is also described as a simple software-level technique that can be used in conjunction with BTI-Gater to avoid certain pathological aging scenarios. Our experiments show that BTI-Gater can balance the gated clock branches to close to 50% signal duty ratio, while guaranteeing a glitch-free clock signal with easy-to-verify timing constraints. Results on commercial processors show that BTI-Gater can effectively reduce N/PBTI-induced clock skew of up to 17ps, which can be converted to up to 19.7% leakage power saving compared to pure design guardbanding.

## I. INTRODUCTION

Negative- and Positive-Biased Temperature Instability (N/PBTI) have become a major reliability concern for modern semiconductor technology [18], [9]. NBTI manifests itself as an increase in PMOS threshold voltage  $V_{th}$  when PMOS is negatively biased, while PBTI manifests as an increase in NMOS  $V_{th}$  if NMOS is positively biased. When the biased voltages are removed, the devices enter a recovery phase, where part of the degradation can be recovered.

Due to this stress-recovery behavior, N/PBTI-induced degradation depends heavily on the workload. Mintarno et. al. [15] report that N/PBTI-induced timing degradation on data path varies from 2% to 11%, depending on the workload. Compared to data path, clock distribution network usually has balanced structure and invariant signal pattern, which makes it more robust against N/PBTI-induced degradation. But this may not hold for clock distribution networks with clock gating features [13]. Conventionally, clock signal on the gated branch

L. Lai and P. Gupta are with the Electrical Engineering Department, University of California Los Angeles, Los Angeles, CA 90095 USA. e-mail: (liangzhen@ucla.edu; puneet@ee.ucla.edu).

V. Chandra and R. Aitken are with ARM Inc., San Jose, CA 95134 USA. e-mail: (Vikas.Chandra@arm.com; Rob.Aitken@arm.com).

This work is supported in part by NSF Variability Expedition grant CCF-1029030.

Copyright ©2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

is frozen at one state (i.e., logic low or logic high), which will introduce imbalanced degradation and cause additional clock skew.

Previous work [6], [10], [7], [5] has attempted to address this issue. However, most of existing work did not consider the clock gating control signal generation and the synchronous nature of the clock gating operations, which may lead to incorrect assumptions and circuit malfunctions. We will review each of them in Section VI.

In this work, we first analyze the effects of N/PBTI on clock distribution network and N/PBTI-induced clock skew under different clock gating use cases. Then cross-layer solutions are proposed to address this issue of N/PBTI-induced clock skew. Since the clock skew is mainly due to the imbalanced clock signal duty ratio ( $D$ ), two ICG cell circuits are proposed to alternate clock idle state between logic high and logic low for each clock gating operation, so that the clock signal duty ratio ( $D$ ) can be balanced to close to 50%. A skew mitigation methodology is also proposed to apply the appropriate ICG cells for different design, based on the architecture and microarchitecture context. To avoid certain pathological aging scenarios, an example of sleep scheduling is also described as a simple software-level technique that can be used in conjunction with BTI-Gater.

The rest of the paper is organized as follows: Section II introduces some background information and discusses the effects of N/PBTI on clock distribution network. Section III analyzes the N/PBTI-induced clock skew for clock distribution network with clock gating. Section IV presents our cross-layer solutions, including two BTI-Gater cells and a skew mitigation methodology and an example of software sleep scheduling. Experimental results are described in Section V. Related work is reviewed in Section VI, and Section VII concludes the paper.

## II. EFFECTS OF N/PBTI ON CLOCK DISTRIBUTION NETWORK

In this section, we present background information and discuss how N/PBTI affects clock distribution network. We first introduce the N/PBTI mechanism and modeling of the device in Section II-A. Then we analyze the effect of NBTI and PBTI on clock path delay in Section II-B.

### A. N/PBTI Mechanism and Modeling

The exact mechanisms of N/PBTI are still under debate within the reliability community. There are two major NBTI aging models: the reaction-diffusion (RD) model [4], [21] and the trapping/detrapping (TD) model [20].

Regardless of the argument of N/PBTI mechanisms and models, it is widely-accepted and well-understood that stress duty ratio is a strong determining factor for the degradation due to its stress-recovery behavior. This work aims at reducing N/PBTI-induced clock skew by balancing the stress-recovery ratio between different clock branches. The purpose of this work is *not* comparing different aging models. In this work, we will evaluate our methodology using different aging models.

We use the RD model in [4], [21]. The time dependence of  $V_{th}$  degradation in this model follows a power-law model as:

$$\Delta V_{th}(t) = Kt^n + M \quad (1)$$

where the time exponent  $n \sim 0.16$ .

We use the TD model in [20]. The time dependence of  $V_{th}$  degradation in this model follows a logarithm timing dependence as:

$$\Delta V_{th}(t) = L \cdot [A + B \cdot \log(1 + Ct)] \quad (2)$$

To get the realistic amount of N/PBTI degradation, in the experiments, we use a commercial sub-32nm process technology. A calibrated industry aging model (including both NBTI and PBTI, without recovery effect) is used as baseline. The RD model as in Equation (1) and the TD model as in Equation (2) are fitted and scaled to match the degradation under close to DC aging ( $D = 99\%$ ). Based on the simulation results, the amount of PBTI degradation is significant and cannot be ignored, comparing to NBTI.

### B. N/PBTI on Clock Path Delay

A clock path example is shown in Fig. 1. Here we assume that the design uses Flip-Flops (FF) that are triggered by clock rising-edge. So clock skew depends only on the clock path delay associated to this signal transition pattern, i.e., transition from logic state A to logic state B as in Fig. 1.

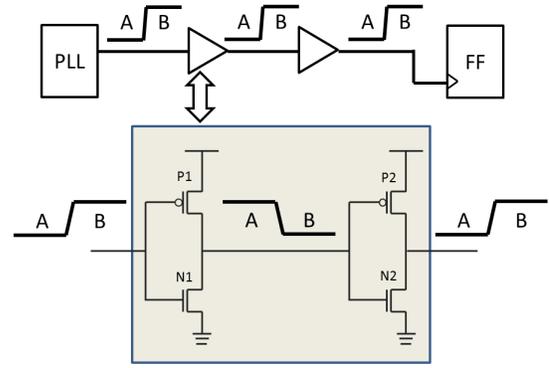
As shown in the table in Fig. 1, at logic state A, device P1 and N2 are stressed while N1 and P2 are relaxed. This degradation pattern will result in weaker (i.e., with larger  $V_{th}$ ) P1 and N2, and stronger (i.e., with smaller  $V_{th}$ ) P2 and N1, all of which will help the signal transition (i.e., reduce the delay) from logic state A to logic state B. The degradation pattern at logic state B is reversed, i.e., stronger P1, N2 and weaker P2 and N1, all of which will slow down the signal transition from logic state A to logic state B.

Therefore, we have two conclusions here:

- 1) Both NBTI and PBTI will result in the same type of clock path delay change. In this work, we will consider both effects of NBTI and PBTI.
- 2) The amount of N/PBTI-induced delay change on the clock path depends on the time it spends on each logic state. In this work, we define the clock signal duty ratio ( $D$ ) as the probability of clock path staying at logic state B as in Fig. 1.

## III. N/PBTI-INDUCED CLOCK SKEW UNDER CLOCK GATING

We have discussed the effects of N/PBTI on clock path delay. In this section, we will explain why N/PBTI-induced



Logic State	N1 status	P1 status	N2 status	P2 status
A	relaxed	stressed	stressed	relaxed
B	stressed	relaxed	relaxed	stressed

Fig. 1. Effects of NBTI and PBTI on clock paths

delay degradation can result in clock skew if clock gating is implemented on the clock distribution network. We first discuss different clock gating use cases in Section III-A. Then we describe the implementation of clock gating using conventional ICG cells in Section III-B. We present the analysis of N/PBTI-induced skew and the vulnerability of different clock gating use cases in Section III-C. Finally, we discuss the impacts of N/PBTI-induced clock skew and clock uncertainty on design timing margining in Section III-D.

### A. Clock Gating Use Cases

Clock gating is one of the most popular power management techniques, due to its low reaction latency and small implementation overhead. It can be implemented at various levels of granularity, depending on the context and use cases. Based on the implementation granularity and software visibility, different clock gating use cases can be classified as follows:

- **Architecture-level clock gating:** The clock signal of the entire processor core or specific module (e.g., co-processor, hardware accelerators etc.) is gated upon the request of the software. This is typically used when turning the processor/module into sleep mode or Wait-For-Interrupt (WFI) mode. The clock gating probability has strong dependency on the actual software workload. An example of architecture-level clock gating is C states in ACPI [1].
- **Microarchitecture-level clock gating:** Many operations in a processor, especially superscalar processor, may be mutually exclusive and utilize different parts of the circuit. Computer architect can identify the idle modules and corresponding clock gating conditions during RTL design phase. Examples include dynamic high level clock gating [2] and deterministic clock gating [12].
- **Circuit-level clock gating:** During the circuit design phase, data dependency information can be derived from the RTL. Based on the data dependency information, clock gating can be inserted automatically [8], [19], [3] without affecting the architectural behavior. This feature is also supported by some commercial tools.

### B. Clock Gating Implementation with ICG Cell

Clock is one of the most important and critical signals in VLSI circuit. Any imperfection in the clock signal may cause circuit malfunction or timing violations. For example, a glitch in clock signal may cause the FF to sample incorrect signal and corrupt the architectural state. Clock skew between launching FF and destination FF can cause timing violations.

Since clock signal integrity is crucial for correct functionality, clock gating is usually implemented using a special ICG cell, which is carefully designed to avoid introducing any suspicious timing behavior or glitches. An example of clock gating implementation using a conventional ICG cell [11] is shown in Fig. 2. The ICG cell consists of one latch and one AND gate. An enable signal (EN) is used to specify whether the output clock (CKOUT) is enabled. If the ICG cell samples the EN signal as logic low, it will gate the output clock and omit the immediate clock pulse (the dotted pulse in Fig. 2) with no clock gating latency.

Although the ICG cell in Fig. 2 uses a latch, its timing behavior is similar to a regular FF, i.e., the signal EN should be and only need to be ready around the incoming clock rising edge with corresponding setup and hold time constraints. This is extremely important for the general applicability of the ICG cell as it specifies the timing requirements of the enable signal EN. In a typical clock gating implementation, the EN signal is generated in the same way as other regular data signals, i.e., driven by the same set of FFs and share the same combinational logic. Therefore, the FF-like timing behavior makes the implementation flow and timing verification consistent with the rest of the design.

Depending on the clock gating granularity, multiple or even thousands of clock sinks can share the same clock gating control. For example, for the architecture-level clock gating use cases, all FFs in the processor core can be gated or ungated together, depending on its sleep state. Usually, only one or a few number of ICG cells will be used at the clock distribution network root. The reasons for this are: 1) this can reduce the number of ICG cells and thus reduce the implementation overhead; 2) this can help saving the clock power dissipation along the clock path. A potential pitfall of this type of implementation is the large clock skew between the ICG cell and its sink FFs (see Fig. 2). For large design, the clock path delay can be comparable or even larger than one clock cycle [17], [16]. In this case, even though the ICG cell can cut off the root clock signal with zero cycle latency, it can still take one or more clock cycles to stop the clock signals at the leaf sinks. Special pipeline arrangement should be used to handle this extra latency, e.g., Power Control Register [2].

### C. Clock Skew Induced by N/PBTI

In a clock distribution network without clock gating, clock signal duty ratio ( $D$ ), i.e., the probabilities of staying at logic state B, is balanced at 50% between different clock path. This results in balanced delay degradation and no additional clock skew. However, if clock gating is implemented, the signal duty ratio  $D$  will be imbalanced between the regular clock branch and gated clock branch, as shown in Fig. 3. In this work, we

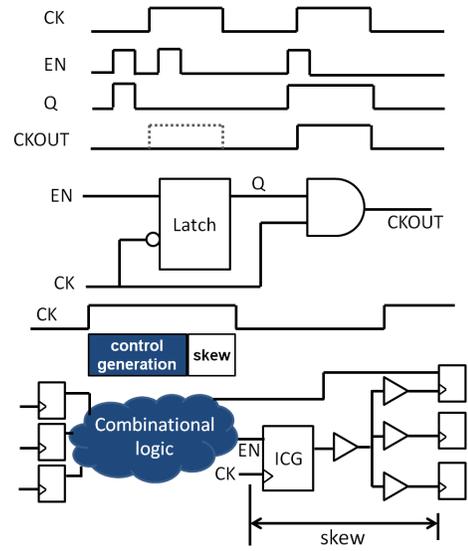


Fig. 2. Conventional ICG cell and its usage in circuits. The timing behavior of the ICG cell is similar to an edge-triggered element.

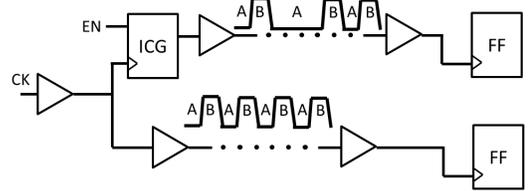


Fig. 3. Imbalanced signal duty ratio ( $D$ ) for gated clock branch and regular clock branch

define the clock activity ratio  $\mu$  as:

$$\mu = 1 - \frac{\text{total clock gated cycles}}{\text{total cycles}} \quad (3)$$

So if conventional ICG cell is used, we have  $D = \mu/2$ .

As shown in earlier analysis in Section II-B, lower  $D$ , or equivalently staying more at logic state A, will result in less degradation, i.e., faster clock paths. Therefore, the gated clock path is expected to become faster than regular clock path after aging if conventional ICG cell is used. Both NBTI and PBTI will result in the same type of clock skew, thus aggravating this effect. The clock skew will be mostly determined by the clock path delay after the ICG cell because the clock signal duty ratio  $D$  is different only for this part. Therefore, we expect architecture-level and microarchitecture-level clock gating to be more vulnerable to N/PBTI-induced clock skew. We will focus on these two types of clock gating use cases in this work. As shown in Table II in Section V-B, the clock skew becomes significant only when the number of gated clock sinks is large enough.

### D. Clock Uncertainty and Design Margining

The exact amount of N/PBTI-induced clock skew depends on the degradation contexts, e.g., lifetime, clock gating activities etc., which are unknown during design time. So it should be considered as additional clock uncertainty on top of other

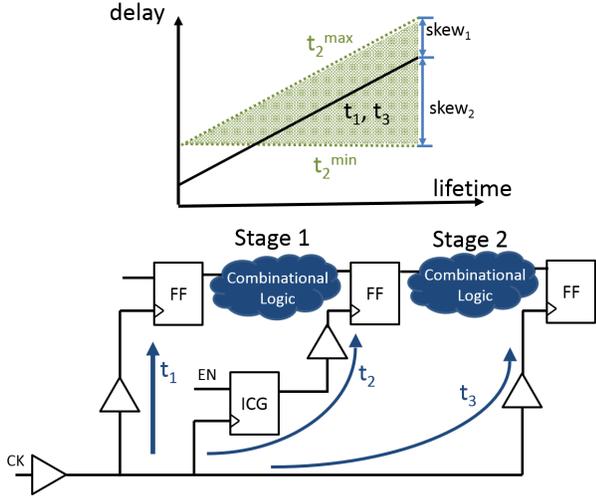


Fig. 4. Sample illustration of N/PBTI-induced clock skew and clock uncertainty.

existing static clock skew. This clock uncertainty can either increase or decrease the clock skew and affect the setup-time or hold-time margin, depending on the relative location of FFs. For example, as shown in Fig. 4,  $skew_1$  will increase the hold-time margin of stage 1 and the setup-time margin of stage 2, while  $skew_2$  will increase the setup-time margin of stage 1 and the hold-time margin of stage 2. Therefore, as long as the total clock uncertainty, i.e.,  $skew_1 + skew_2$ , keeps the same, the total design margin can not be reduced.

Unless we can reduce the total delay uncertainty of gated clock path, it is difficult to reduce the design margin by using static design methods. For example, even though we know that the gated clock path is likely to become faster comparing to regular clock path, a static delay shift for gated clock path at  $t = 0$ , i.e., shifting the shaded part upwards, can only change the split between  $skew_1$  and  $skew_2$  rather than reducing the total clock uncertainty.

So clock gating can change the clock signal duty ratio  $D$  of the gated clock path, resulting in additional clock skew and thus design margin. If the ICG cell can alternate clock idle state between logic state A and logic state B, the clock signal ratio  $D$  can be balanced to close to 50%, which can eliminate the clock skew introduced by N/PBTI. This motivates our proposed clock gating methodology.

#### IV. N/PBTI-RESILIENT CLOCK GATING METHODOLOGY

In this section, we first discuss the challenges and issues in designing ICG cells with controllable idle state in Section IV-A. Then we describe BTI-Gater cells with half cycle and one cycle clock gating latency in Section IV-B and IV-C. A skew mitigation methodology is presented in Section IV-D. Finally, software scheduling techniques are described in Section IV-E to avoid certain pathological scenarios.

##### A. Clock Gating Latency

Theoretically, it is impossible to implement an idle high ICG cell with zero cycle latency. As shown in the example in Fig. 5,

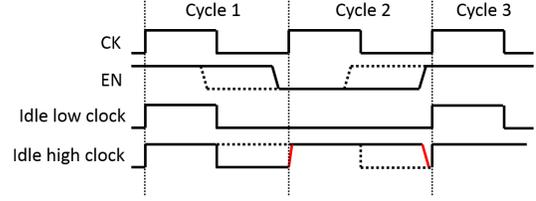


Fig. 5. Illustration of the clock gating latency issue

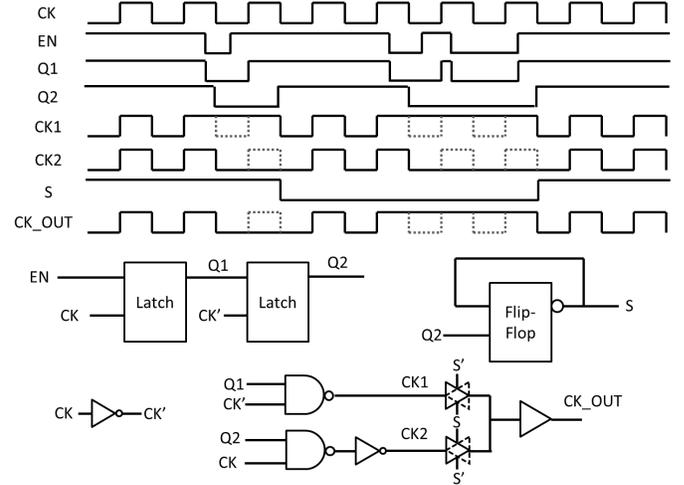


Fig. 6. BTI-Gater cell with half cycle clock gating latency

at the time when correct clock gating control signal arrives at ICG cell, the output clock is already at logic low state (see the end of cycle 1 as in Fig. 5). This makes the AND-type gate, i.e., with logic low idle state, the only option for an ICG cell. If we use an OR-type gate, i.e., making the idle state at logic high, the output clock will inevitably generate an undesired clock rising edge (the red rising edge at the end of cycle 1 as in Fig 5), which can corrupt the architecture states. The same issue happens when enabling the clock, where an additional clock falling edge (the red falling edge at the end of cycle 2) is needed.

The same argument holds if we apply the ICG cell to an inverted clock stage in an inverter-based clock distribution network. The clock control signal, in this case, is coming at the clock falling edge, which makes the OR-type gate the only possible ICG cell with idle state being logic high. Note that at the inverted clock stage, logic high state is equivalent to the logic low state at non-inverted stage. If we use an AND-type gate, the output clock will generate an undesired clock falling edge, which corresponds to a clock rising edge at the FFs.

Therefore, it is impossible to use both AND-type and OR-type gates in ICG cell without changing the clock gating latency. The minimum clock gating latency for an ICG cell with controllable idle state is half clock cycle, i.e., the enable signal arrives before the clock falling edge.

##### B. Proposed BTI-Gater Cell of Half Cycle Latency

For an ICG cell with half cycle latency, the enable signal is sampled at clock falling edge. Our proposed ICG cell

generates two copies of the gated clock signal with different idle states and alternatively select the appropriate one.

The complete design of BTI-Gater cell with half cycle latency is shown in Fig. 6. The signal CK1 is the clock signal with logic high idle state. The signal CK2 is the clock signal with logic low idle state. An internally generated signal S is used to select between the two clock signals CK1 and CK2. The value of S is supposed to change each time the circuit enters clock gating state. This BTI-Gater cell requires the EN signal to be ready at clock falling edge, which leaves about half clock cycle to generate the control signal.

There are several key implementation points here:

- First, we use one master-slave FF to generate both of signal Q1 and signal Q2. This saves the area and reduces the power overhead.
- S is flipped on Q2's rising edge. Since the rising edge of Q2 is synchronous to CK and happens when both CK1 and CK2 are logic low, this implementation makes sure that no additional glitch is introduced when alternating the state.
- The delays in generating CK1 and CK2 are matched to have exactly one NAND gate and one inverter. This can reduce the clock skew mismatch between the idle high and idle low clock output under variations.

### C. Proposed BTI-Gater Cell of One Cycle Latency

As discussed earlier in Section III-B, there are cases where clock gating action can not finish within one clock cycle. Special pipeline arrangement is used to handle this additional latency, e.g., Power Control Register [2]. For this case, an ICG cell with one cycle latency can be accepted and even preferred, because its timing verification is consistent with the rest of the circuits.

We propose BTI-Gater cell with one cycle latency as shown in Fig. 7. The design is similar to BTI-Gater cell with half cycle latency. One more latch is inserted to buffer the enable signal. The same key implementation consideration is applied as well. As shown in the timing diagram in Fig. 7, BTI-Gater cell will block the second rising edge, i.e., with one cycle latency, after detecting the enable signal.

### D. Mitigation Methodology

Depending on the architectural context and latency requirements, different ICG cells may be used. For example, some microarchitecture-level clock gating cases may specifically require sub-cycle latency. So only BTI-Gater cell of half cycle latency can be used. If sub-cycle latency is not required, BTI-Gater cell of one cycle latency may be preferred because of its simpler and more consistent delay constraints (with respect to the clock rising edge only).

Based on the properties of the two BTI-Gater cells, we propose a skew mitigation methodology to select the appropriate ICG cell, depending on the architectural context and clock gating latency requirements. The methodology is described in Fig. 8. First, the clock gating use cases and corresponding latency requirements are identified. Corresponding candidate

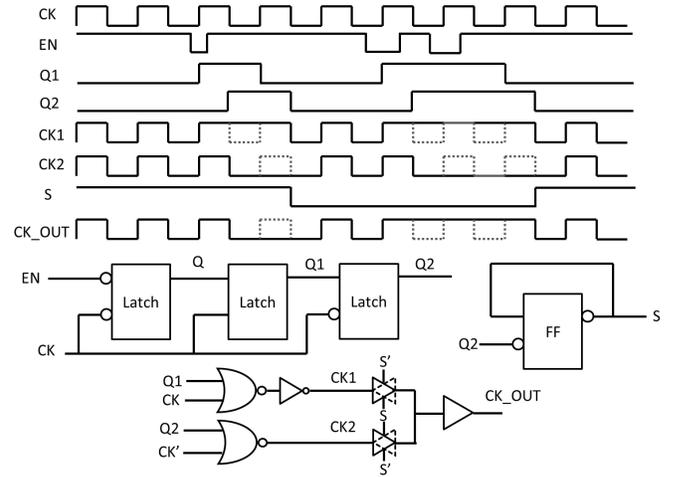


Fig. 7. BTI-Gater cell with one cycle clock gating latency

BTI-Gater cells are selected. For clock gating cases with sub-cycle latency and more than half cycle control signal generation time, design guardbanding is the only option. For other cases, the cost efficiency can be calculated and used to determine if it is beneficial to use BTI-Gater cells. The power benefit of using BTI-Gater cells can be modeled by a sample cost function as Equation (4):

$$\Delta P = \Delta t_{skew}(\alpha \cdot p_s + \beta \cdot p_h) - P_{cell} \quad (4)$$

where  $\Delta t_{skew}$  is the skew saving of using BTI-Gater cells,  $p_s$  is the power cost to margin for per unit setup time uncertainty,  $p_h$  is the power cost to margin for per unit hold time uncertainty,  $P_{cell}$  is the additional power cost of using BTI-Gater cells over conventional ICG cells,  $\alpha$  is the percentage of circuits being setup time constrained,  $\beta$  is the percentage of circuits being hold time constrained. In practice, the value of  $P_{cell}$  can be pre-characterized.  $\Delta t_{skew}$  can be derived through circuit simulation on the clock path using typical and aged libraries. Direct derivation of  $p_s$ ,  $p_h$ ,  $\alpha$ , and  $\beta$  is difficult. But the entire term  $\Delta t_{skew}(\alpha \cdot p_s + \beta \cdot p_h)$  can be estimated through design optimization tools by adding/removing  $\Delta t_{skew}$  to/from the total clock uncertainty<sup>1</sup>. Design intuition based on the same or similar design/technology can also be utilized to estimate the cost of additional setup and hold margin (i.e.,  $p_s$  and  $p_h$ ).  $\Delta t_{skew} \cdot \alpha$  and  $\Delta t_{skew} \cdot \beta$  can be derived using Total Negative Slack (TNS), which is supported by most design tools.

### E. Software Sleep Scheduling Techniques

N/PBTI degradation on datapath will result in slower path delay and lead to setup-time violations. System adaptation schemes [14] can be applied to reduce the design guardband. However, N/PBTI degradation on clock path can result in hold-time violations, which are difficult to resolve through system adaptation schemes. Therefore, from hardware's point of view, design margin can not be reduced as long as there are certain

<sup>1</sup>In this work, we apply Vt-assignment using commercial tools to estimate the power saving (see Section V-E and Fig. 12).

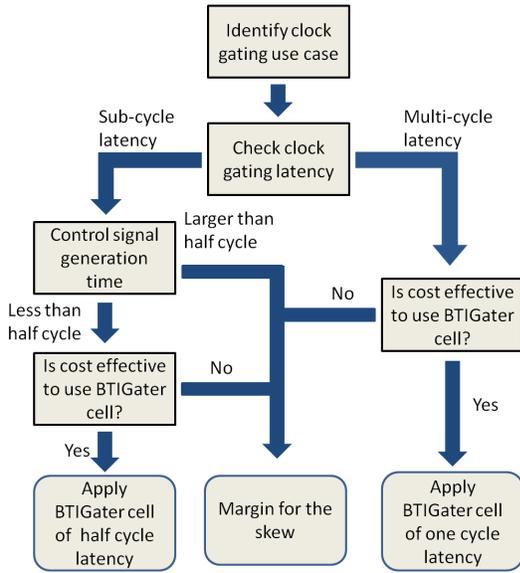


Fig. 8. N/PBTI-induced clock skew mitigation methodology

pathological scenarios that can lead to imbalanced degradation, especially for architecture-level clock gating, where the software can have very regular behavior.

For all typical software workload experiments we ran (see Table III in Section V), BTI-Gater can balance the clock signal duty ratio ( $D$ ) to close to 50%. But one may still be able to construct certain pathological cases that can result in highly imbalanced degradation even with BTI-Gater. For example, software may have two program phases where the first phase always results in a very short period of clock gating time while the second one always results in a very long period of clock gating time.

To avoid these pathological scenarios, software sleep scheduling techniques in conjunction with BTI-Gater can be applied for architecture-level clock gating cases to balance the clock degradation. A software wrapper can be implemented on the sleep function. A sample software wrapper pseudo code is shown in Algorithm 1. A software bit  $idle\_state$  is set to keep track of the previous clock gating status. Even status and odd status correspond to the idle state low and idle state high cases of BTI-Gater cells. Based on  $idle\_state$ , the wrapper uses a software counter  $counter$  to keep track of the difference between the sleep time spent in even and odd clock gating operations. Upon receiving the sleep request, the wrapper will first determine the desired sleep state based on the value of  $counter$  and  $idle\_state$ . The sleep operation is broken into two parts ( $time/3$  and  $time*2/3$ ) and scheduled accordingly to achieve a total of  $time/3$  spent in the desired sleep state. This makes sure that the absolute value of  $counter$  is bounded by one third of the maximum value of  $time$ , if this sleep function is the only mechanism for clock gating. This is tiny and negligible compared to typical hardware lifetime of several years.

If clock-gating is invoked when the processor receiving a WFI request, slightly modified software scheduling techniques can be applied. Instead of breaking the sleep time into two

parts, the wrapper can enforce a short period of sleep before turning into WFI state in order to balance the value of  $counter$ . In this case, the absolute value of  $counter$  is bounded by the maximum sleep time. If the system expected sleep time can be large and non-negligible compared to its lifetime, a periodic wake-up mechanism can also be used to limit the maximum time of a single sleep operation.

---

#### Algorithm 1 Software wrapper for sleep scheduling

---

**Require:**  $idle\_state$  is previous clock gating status.  $counter$  is the software counters to record the difference between the sleep time spent in even and odd clock gating operations.

```

function MYSLEEP( $time$ )
  if  $counter > 0$  then
    //More time spent in even clock gating status
    if  $idle\_state == \text{even}$  then
      SLEEP( $time*2/3$ )
      SLEEP( $time/3$ )
       $counter -= time/3$ 
       $idle\_state = \text{even}$ 
    else
      //Force the clock idle state
      SLEEP( $time/3$ )
      SLEEP( $time*2/3$ )
       $counter -= time/3$ 
       $idle\_state = \text{odd}$ 
    end if
  else
    //More time spent in odd clock gating status
    if  $idle\_state == \text{odd}$  then
      SLEEP( $time*2/3$ )
      SLEEP( $time/3$ )
       $counter += time/3$ 
       $idle\_state = \text{odd}$ 
    else
      //Force the clock idle state
      SLEEP( $time/3$ )
      SLEEP( $time*2/3$ )
       $counter += time/3$ 
       $idle\_state = \text{even}$ 
    end if
  end if
end function
  
```

---

## V. EXPERIMENTAL RESULTS

In this section, we first describe our experiment setup in Section V-A. The clock skew analysis results are presented in Section V-B, which verify our earlier analysis in Section III-C. Section V-C reports the results on BTI-Gater cells, including functionality verification, area and power overhead, and clock jitter evaluation. Two different case studies using BTI-Gater cells are performed in Section V-D and Section V-E.

### A. Experiment Setup

In this work, we perform clock gating case study on two commercial processors. Both processors have built-in micro-

architecture-level clock gating options. Processor B has built-in architecture-level clock gating to gate part of its peripheral when idle. We also consider the architecture-level clock gating cases when the entire processor core pipeline is clock gated when receiving a sleep request. The different clock gating cases are summarized in Table. I.

To analyze N/PBTI-induced clock skew, we use a commercial sub-32nm process technology and libraries. An industry calibrated aging model (including both NBTI and PBTI, without recovery effect) is used as the baseline. To include the recovery effect, both RD and TD models are fitted and scaled to match the technology node, desired operating conditions and lifetime requirements. We see about 10% lifetime delay degradation for clock buffer chain under 100% stress duty ratio.

### B. N/PBTI-Induced Skew Analysis

In this analysis, we evaluate the amount of N/PBTI-induced clock skew for each of the clock gating cases, assuming only conventional ICG cells are used. For each of the clock gating cases, we first extract the corresponding clock path after the ICG cells. Then we perform circuit simulation on these paths with degradation based on different aging models and different clock activity ratios ( $\mu$ ). The results are highlighted in Table II. The 100% clock activity ratio represents the scenario that the clock path is never gated. The heavily gated scenario, i.e., with smaller  $A$ , has relatively smaller clock path delay, which matches our earlier analysis in Section III-C.

The N/PBTI-induced clock skew equals the delta between ungated cases ( $A = 100\%$ ) and corresponding entries in Table II. The skew when  $A = 2\%$  is also listed in Table II. The amount of N/PBTI-induced skew depends heavily on the clock path delay. For the longest path, i.e., Case I, the maximum N/PBTI-induced skew can be as much as 17ps based on TD model. While for short paths, e.g., Case III, IV and V, the N/PBTI-induced skew is very small.

### C. BTI-Gater Cell Results

To validate BTI-Gater cell design, we implement both cells using the same sub-32nm technology and libraries. Their functionality is verified by SPICE simulation and is consistent with the timing diagrams in Fig. 6 and Fig. 7. The area of half cycle latency BTI-Gater cell is equivalent to about 25 standard size inverters or 4 standard size FF. The area of one cycle latency BTI-Gater cell is equivalent to about 31 standard size inverters or 5 standard size FF. The area of

TABLE I  
SUMMARY OF DIFFERENT CLOCK GATING CASES

Case	Benchmark	Gated Module	Clock sinks	Clock levels
I	Processor A	Entire core pipeline	~ 30k	9
II	Processor A	Part of core pipeline	~ 10k	8
III	Processor B	Entire core pipeline	~ 1k	4
IV	Processor B	Part of peripherals	~ 100	2
V	Processor B	Part of core pipeline	~ 350	2

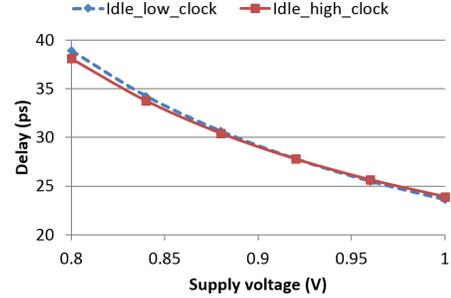


Fig. 9. Clock delay under different supply voltages. The clock jitter introduced by BTI-Gater is very small.

conventional ICG cell is equivalent to about 5 standard size inverters. The power consumption of our proposed ICG cells is about 4X larger than conventional ICG cell. However, in all studied cases, the processor requires at most four ICG cells to implement the clock gating functions. Compared to the size of the entire design and its clock distribution network (see Table I), the power and area overhead of using BTI-Gater is almost negligible. For cases with small clock network, e.g., Case IV and V, the proposed methodology as shown in Fig. 8 can be applied to decide whether it is beneficial to use BTI-Gater cells.

SPICE simulations are also used to evaluate the potential clock jitter introduced by BTI-Gater cell (see Fig. 9). The jitter is within 1ps across supply voltage ranging from 0.8V to 1V, which is small compared to the clock uncertainty introduced by local process variations.

### D. Case Study for BTI-Gater Applicability

To verify and demonstrate the applicability of BTI-Gater, we apply BTI-Gater methodology for Case V. Since Case V specifically requires sub-cycle clock gating latency, BTI-Gater cell of half cycle latency is considered. The processor is synthesized, placed and routed with the clock gating implemented using conventional ICG cell. After verifying the clock gating control signal generation time, we replace it with BTI-Gater cell of half cycle latency through Engineering Change Order (ECO). The design is verified again after the ECO to make sure that the clock gating control signal meets the timing constraints with respect to the clock falling edge. The layout with clock distribution networks highlighted is shown in Fig. 10.

We also perform functional simulation of processor A by running Dhrystone [22] benchmark. The clock activity ratio  $\mu$  in Case V is less than 9%, which implies a clock signal duty ratio ( $D$ ) of less than 5% for the gated clock path if conventional ICG cell is used. Our BTI-Gater cell is able to generate a clock signal with  $D = 52.68\%$ , which is very close to standard clock signal duty ratio of 50%.

### E. Case Study for Power Saving

To evaluate the benefit and potential power saving of our skew mitigation methodology, we apply BTI-Gater methodology on Case I. Since Case I is architecture-level clock gating,

TABLE II  
CLOCK PATH DELAY AND SKEW (IN PS) UNDER DIFFERENT AGING MODELS AND ACTIVITIES USING CONVENTIONAL ICG CELLS

	Fresh (at t=0)	$\mu = 100\%$ (i.e., un gated)			$\mu = 50\%$			$\mu = 2\%$			Skew when $\mu = 2\%$		
		RD	TD	Industry	RD	TD	Industry	RD	TD	Industry	RD	TD	Industry
Case I	311.4	328.9	328.9	342.7	326.6	320.3	339.6	320.7	312.1	311.2	8.1	16.8	31.5
Case II	229.1	240.8	239.6	250.7	239.0	234.0	248.5	234.8	228.7	228.0	6.0	10.9	22.7
Case III	90.1	93.7	93.4	96.6	93.2	91.8	95.9	91.9	90.2	90.0	1.8	3.2	6.6
Case IV	47.2	47.2	47.2	50.6	47.2	47.2	50.3	47.2	47.2	47.2	0.0	0.0	3.4
Case V	41.5	41.5	41.5	44.2	41.5	41.5	43.9	41.4	41.4	41.5	0.1	0.1	2.7

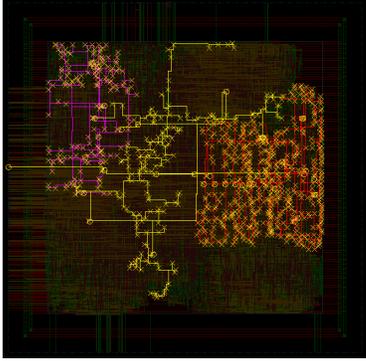


Fig. 10. Layout for processor B with clock distribution networks highlighted (ungated part in yellow, Case IV in purple and Case V in red)

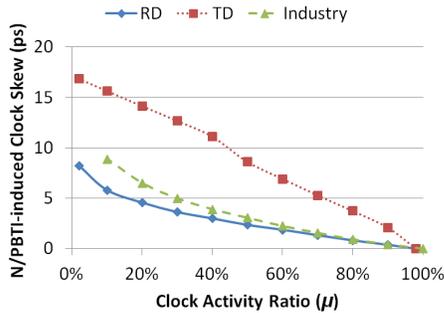


Fig. 11. Clock skew with conventional ICG cell for Case I

BTI-Gater cell of one cycle latency is used. The corresponding clock skew with conventional ICG cell under different clock activity ratio ( $\mu$ ) is shown in Fig 11. The maximum skew is about 17 ps with TD model and about 10 ps for other aging models. We also expect the clock gating probability to be a strong function of the software workload. We perform simulation of various software benchmarks and generate the active-sleep patterns. The benchmark active-sleep patterns are used to simulate the clock signal duty ratio  $D$  for both conventional ICG cell and BTI-Gater cells. As shown in Table III, BTI-Gater cells can balance the clock signal duty ratio  $D$  to the range between 46.73% and 56.53%, which will reduce the corresponding clock skew to less than 1ps. This improvement is expected to be larger if more software iterations are running on one processor or the software scheduling techniques as in Section IV-E is used.

To evaluate the power saving of the clock skew reduction, we make the following assumptions: 1) the designer will

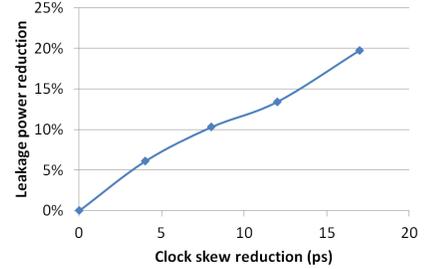


Fig. 12. Leakage power saving by Vt-assignment with different clock reduction for Case I

TABLE III  
CLOCK ACTIVITY RATIO( $\mu$ ) AND SIGNAL DUTY RATIO( $D$ )  
RESULTS FOR DIFFERENT SOFTWARE BENCHMARKS

Benchmark	Clock activity ratio ( $\mu$ )	Clock signal duty ratio ( $D$ )	
		Conventional	BTI-Gater
mp3	2.14%	1.07%	49.98%
Dhrystone	39.26%	19.63%	56.53%
3D rendering	23.25%	11.62%	46.73%
Web browsing	6.78%	3.39%	52.66%

guardband for the worst-case P/NBTI-induced skew if conventional ICG cell is used; 2) BTI-Gater cell can balance the clock signal duty ratio  $D$  to around 50%. In the experiment, this processor design is first synthesized, placed, routed, and signed-off with the maximum N/PBTI-induced skew (i.e., 17ps) as additional clock uncertainty margin. Then we apply BTI-Gater methodology and remove the corresponding amount of clock uncertainty. The additional timing slack is reclaimed by Vt-assignment using commercial tools. The leakage power results for different amounts of skew reduction are plotted in Fig. 12. The results show that our method can save up to 19.7% leakage power from the 17ps relaxation in clock uncertainty. Since Vt-assignment will only swap cells with the same gate size but different Vt options, the dynamic power remains the same.

## VI. RELATED WORK

There is several related work aiming at reducing N/PBTI-induced clock skew issue by different approaches.

Chakraborty et. al. [6] proposed a scheme to statically select AND-type or OR-type cells as the output stage of the ICG cell. There are three potential drawbacks in this method. First, the method relies on knowing the clock gating probability at design time. With incorrectly assumed clock

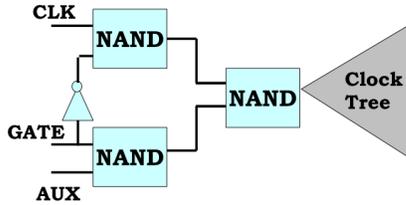


Fig. 13. The ICG cell proposed in [5].

gating probability, the method may aggravate rather than reduce N/PBTI-induced skew. Second, the method did not consider the generation of clock control signal, which leads to the inappropriate assumption that both AND-type and OR-type ICG cells can be used at the same time. As discussed in Section IV-A, it is non-trivial to implement both AND-type and OR-type ICG cells. Last, the method works only if clock gating is implemented at multiple locations in the clock distribution network, which limits its applicability.

Huang et. al. [10] proposed a scheme to identify critical PMOS in the clock distribution network and selectively use AND-type or NAND-type ICG cells. This method focuses on reducing the aging effect within the ICG cell rather than the clock buffers after it. Therefore, this method can only offer limited benefit in reducing N/PBTI-induced clock skew for clock gating with coarse granularity, which is more vulnerable to N/PBTI-induced clock skew.

Chen et. al. [7] proposed a scheme to selectively replace cells on clock path with higher  $V_t$  cells to compensate both aging and process variation. This method applies  $V_t$ -assignment on the clock distribution network so that the clock path with larger degradation has smaller initial delay at  $t=0$ . However, as discussed in Section III-D, this method may not reduce the clock uncertainty due to N/PBTI-induced skew, thus may not reduce the design margin.

Chakraborty et. al. [5] proposed an ICG cell design (see Fig. 13) that can alternate the clock state based on an external signal AUX. This is the most closely related work. However, this method does not consider the synchronization of the control and state alternating signals. This leads to the same assumption that AND-type and OR-type gating cells can be used at the same time, which is non-trivial to implement.

## VII. CONCLUSIONS

In this paper, we first analyzed the effect of N/PBTI on clock distribution network with clock gating features. Then we proposed two BTI-Gater cells that can be used to balance delay degradation on gated clock branch and regular clock branch. Last we proposed an N/PBTI-induced skew mitigation methodology. Software sleep scheduling techniques are described in conjunction with BTI-Gater to avoid certain pathological aging scenarios. Experimental results show that BTI-Gater cells can be used to reduce N/PBTI-induced clock skew by up to 17ps. By using our skew mitigation methodology, we can save up to 19.7% leakage power compared to pure design guardbanding. Our future work includes a silicon prototype of the proposed approach and its validation under accelerated aging scenarios.

## REFERENCES

- [1] Advanced configuration and power interface specification. <http://acpi.info/spec.htm>.
- [2] ARM Cortex-A9 technical reference manual. <http://www.arm.com/products/processors/cortex-a/cortex-a9.php>.
- [3] L. Benini et al. Automatic synthesis of low-power gated-clock finite-state machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(6):630–643, 1996.
- [4] S. Bhardwaj et al. Predictive modeling of the nbtii effect for reliable design. In *IEEE Custom Integrated Circuits Conference*, pages 189–192. IEEE, 2006.
- [5] A. Chakraborty et al. Analysis and optimization of nbtii induced clock skew in gated clock trees. In *IEEE/ACM Design, Automation and Test in Europe*, pages 296–299, 2009.
- [6] A. Chakraborty et al. Skew management of nbtii impacted gated clock trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6):918–927, 2013.
- [7] J. Chen et al. A novel flow for reducing clock skew considering nbtii effect and process variations. In *IEEE International Symposium on Quality Electronic Design*, pages 327–334, 2013.
- [8] J. Cong et al. Behavior-level observability analysis for operation gating in low-power behavioral synthesis. *ACM Transactions on Design Automation of Electronic Systems*, 16(1):4, 2010.
- [9] J. Hicks et al. 45nm transistor reliability. *Intel Technology Journal*, 12(2):131–144, 2008.
- [10] S.-H. Huang et al. Low-power anti-aging zero skew clock gating. *ACM Transactions on Design Automation of Electronic Systems*, 18(2):27, 2013.
- [11] M. Keating et al. *Low Power Methodology Manual: For System on Chip Design*. Springer, 2007.
- [12] H. Li et al. Deterministic clock gating for microprocessor power reduction. In *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pages 113–122. IEEE, 2003.
- [13] W. Liu et al. Nbtii effects on tree-like clock distribution networks. In *Proceedings of the great lakes symposium on VLSI*, pages 279–282. ACM, 2012.
- [14] E. Mintarno et al. Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(5):760–773, 2011.
- [15] E. Mintarno et al. Workload dependent nbtii and pbtii analysis for a sub-45nm commercial microprocessor. In *IEEE International Reliability Physics Symposium (IRPS)*, pages 3A.1.1–3A.1.6, 2013.
- [16] P. Restle et al. Timing uncertainty measurements on the power5 microprocessor. In *Proc. IEEE International Solid State Circuits Conference*, pages 354–355. IEEE, 2004.
- [17] P. J. Restle et al. A clock distribution network for microprocessors. *IEEE Journal of Solid State Circuits*, 36(5):792–799, 2001.
- [18] D. K. Schroder et al. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1–18, 2003.
- [19] G. E. Téllez et al. Activity-driven clock design for low power circuits. In *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pages 62–65. IEEE Computer Society, 1995.
- [20] J. Velamala et al. Statistical aging under dynamic voltage scaling: A logarithmic model approach. In *IEEE Custom Integrated Circuits Conference*, pages 1–4, 2012.
- [21] W. Wang et al. An integrated modeling paradigm of circuit reliability for 65nm cmos technology. In *IEEE Custom Integrated Circuits Conference*, pages 511–514. IEEE, 2007.
- [22] R. P. Weicker. Dhrystone: a synthetic systems programming benchmark. *Commun. ACM*, Oct. 1984.