TOOLS AND TECHNIQUES FOR LEAKAGE POWER ANALYSIS

A Thesis in
Computer Science and Engineering
by
Yuh-Fang Tsai

© 2005 Yuh-Fang Tsai

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2005
The thesis of Yuh-Fang Tsai was reviewed and approved* by the following:

Vijaykrishnan Narayanan  
Associate Professor of Computer Science and Engineering  
Thesis Advisor, Chair of Committee

Mary Jane Irwin  
A. Robert Noll Chair of Engineering

Yuan Xie  
Assistant Professor of Computer Science and Engineering

W. Kenneth Jenkins  
Professor of Electrical Engineering

David Duarte  
Analog Design Engineer  
Intel  
Hillsboro, Oregon  
Special Member

Acharya Raj  
Professor of Computer Science and Engineering  
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

Increasing leakage current, which is the current flowing through a transistor while not switching, is one of the major concerns for current and future technologies. A comprehensive analysis of the run-time leakage reduction techniques for both subthreshold leakage and gate leakage is presented. The techniques evaluated include input vector control, body bias control and power supply gating. The impacts of technology scaling on leakage reduction techniques are also evaluated. Due to the lack of leakage reduction technique for interconnects, new schemes for leakage-aware crossbar designs are also proposed. In order to facilitate exploration of leakage power trade-offs at a higher level of abstraction, an architectural level leakage power estimation tool which is built on top of a cycle accurate simulator is proposed. The tool accounts for the design styles, process variations, and the leakage power is evaluated at run-time using dynamic temperature feedback.

Process variations (such as variations in transistor channel length and transistor threshold voltage) as a percentage of their nominal values increase when technology advances. Many of the theories behind leakage reduction techniques are based on controlling these parameters which vary increasingly across technology scaling. Thus, the interrelationship between leakage reduction mechanisms and process-related delay and leakage uncertainty is investigated. The results presented point out the importance of considering the delay and leakage uncertainties when applying leakage reduction techniques and quantify the uncertainties caused by process variations.
# Table of Contents

List of Figures viii

List of Tables xi

Chapter 1

Introduction 1

1.1 Introduction ......................................... 1

1.2 Sources of Leakage Current ............................. 2

1.2.1 Subthreshold leakage .................................. 3

1.2.2 Gate leakage .......................................... 5

1.2.3 Band-to-band tunneling current ...................... 5

1.3 Influence of variations in process and operating conditions .... 6

1.4 Overview of Thesis .................................... 7

1.5 Contributions of This Thesis ............................ 9

Chapter 2

Analysis of Leakage Power Reduction Techniques 10

2.1 Introduction ........................................... 10

2.2 Run-Time Leakage Reduction Techniques .................. 12

2.2.1 By input vector control ............................. 12

2.2.2 By increasing the threshold voltage ................. 12

2.2.3 By gating the supply voltage ....................... 13

2.3 Experimental Setup .................................... 13

2.3.1 Input vector control ................................. 14

2.3.1.1 Datapath logic ................................ 14

2.3.1.2 Memory structures ............................ 15

2.3.1.3 Interconnects ................................. 16
## List of Figures

1.1 The trend of process power. The curve with diamond shows dynamic power while that with triangle shows leakage power. [2] .......... 2  
1.2 The major components of leakage current in a NMOS transistor. ... 3  
1.3 Subthreshold leakage as a function of gate length. [5] ............. 4  
1.4 Gate leakage as a function of gate oxide thickness. [5]. ........... 6  
1.5 Junction leakage as a function of channel doping concentration. subthreshold leakage and gate leakage at 30nm technology are also shown for comparison.[5]. .......................... 7  
1.6 Impacts from temperature and process variations on subthreshold and gate leakage. ................................................. 8  
2.1 Modified low-leakage latches with optimum sleep values stored (1 left, 0 right). .......................................................... 15  
2.2 The PLL as a voltage regulator. ........................................ 17  
2.3 Representation of leakage power behavior during scheme application. 19  
3.1 The architecture of interconnect in network-on-chip. F is the flit size in bits. There are F bits in each input/output port. .......... 32  
3.2 Leakage breakdown in a port of crossbar. .............................. 32  
3.3 Schematic of output to PE direction path of DFC. ................. 33  
3.4 Schematic of output to PE path of DPC design. Both of pre-charged to HIGH and pre-charge to low are shown. .................. 35  
3.5 Logic of pre-charge signal and timing of DFC and DPC. Note that pre-charge are deactivated when in idle. ....................... 36  
3.6 Segmented dual-V_{th} feedback crossbar design. ................... 37  
3.7 Segmented dual-V_{th} pre-charge crossbar design. .................. 38
3.8 Maximum delay, average delay and energy-delay product versus the flit size. Note that SC fails to meet 1GHz timing requirement for 1024-bits flit size. .................................................. 43

4.1 The power simulator framework. ........................................ 46
4.2 Floor plan assumed based Itanium architecture. .................. 48
4.3 3-level model hierarchy. ................................................ 49
4.4 Comparison between results from model and Hspice simulations. 52
4.5 Leakage components at different bias conditions. ............... 53
4.6 Leakage analysis of each structure component. .................... 54
4.7 Leakage power of components been analyzed at circuit level. Both model and Hspice simulation results are shown. ............. 55
4.8 Comparison between the results from model and HSPICE simulations at microarchitecture level. ................................. 59
4.9 Leakage breakdown of benchmarks. ................................... 61
4.10 Dynamic/Leakage power breakdown with temperature feedback. 62
4.11 Leakage power breakdown without process variations(left), and that with process variations (right). Result of benchmark:tomcatv. 62
4.12 Subthreshold and gate leakage breakdown of major components. (Benchmark:132.ijpeg) ............................................. 63

5.1 Leakage power distribution for 65nm and 45nm technologies. .... 66
5.2 A branched inverter chain as the test circuit. ....................... 70
5.3 Design trade-off for optimized gate length. ............................ 71
5.4 \(V_{dd}/V_{th}\) optimization for dynamic power. \(V_{dd}\) is from 0.7V to 1.2V and \(V_{th}\) at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case \(V_{dd}/V_{th}=1.0V/0.2V\). .................................................. 72
5.5 \(V_{dd}/V_{th}\) optimization for leakage power. \(V_{dd}\) is from 0.7V to 1.2V and \(V_{th}\) at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case \(V_{dd}/V_{th}=1.0V/0.2V\). .................................................. 72
5.6 \(V_{dd}/V_{th}\) optimization for leakage power uncertainty. \(V_{dd}\) is from 0.7V to 1.2V and \(V_{th}\) at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case \(V_{dd}/V_{th}=1.0V/0.2V\). .................................................. 73
5.7 \(V_{dd}/V_{th}\) optimization for delay. \(V_{dd}\) is from 0.7V to 1.2V and \(V_{th}\) at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case \(V_{dd}/V_{th}=1.0V/0.2V\). 73
5.8 $V_{dd}/V_{th}$ optimization for delay uncertainty. $V_{dd}$ is from 0.7V to 1.2V and $V_{th}$ at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case $V_{dd}/V_{th}=1.0V/0.2V$. ........................................ 74
5.9 Leakage distribution of evaluated techniques. .................. 76
5.10 Delay time of test circuit for each technique. W point is the worst-case corner delay of Orig. ................................. 78
5.11 Temperature sensitivity of uncertainties of the evaluated techniques. ................................. 80
5.12 Illustration of delay time vs gate driving voltage. ......... 81
List of Tables


2.2 Summary of simulation conditions. ................................................. 14

2.3 Optimized substrate bias level for the units evaluated. ........ 17


2.5 Various performance parameters of LBB on a 32x4 SRAM array. ............................................. 20

2.6 Various performance parameters of IVC on a 8mm bus with staggered buffers. The performance penalty is less than 1 cycle. ................. 21


2.8 Performance penalty and delay time for transitions between sleep mode and active mode and area penalty for BBC. .......................... 24


2.11 Power reduction and overhead of Gated-\(V_{dd}\) applied to a 128-bit SRAM array. P: PMOS, N: NMOS, C: CMOS sleep transistor.


2.14 Comparison of impacts of technology scaling. The ratio in the minimum idle time is the ratio of the minimum idle time in cycles in 0.18\(\mu\m\) to that in 70nm. Cycle time scaled by 0.7x per generation.

3.1 Summary of simulation conditions.

3.2 Summary of evaluated crossbar schemes.

3.3 Summary of simulation results for evaluated crossbar schemes. The power consumptions for pre-charge schemes are obtained by assuming 50% static probability which is the worst case for power. The savings and penalty are the results compared against results of SC.

3.4 The comparison of leakage savings of each scheme with and without process variations. PV: process variation.

4.1 The major parameters and current measurements of the 65nm device model used. Measurement is done at \(V_{dd}=1V\) and 100\(^{\circ}\)C.

4.2 Circuit-level leakage analysis of an inverter.

4.3 Architecture configurations assumed.

4.4 Benchmarks used in experiments.
5.1 Simulation conditions. ........................................... 70
5.2 Leakage reduction techniques evaluated. ....................... 75
5.3 Simulation results of 65nm and 45nm technologies. .......... 75
5.4 Results in 65nm technology. .................................... 77
5.5 Results in 45nm technology. .................................... 77
5.6 Recommended target designs for each technique. ............ 79
1.1 Introduction

For the past 50 years, Metal Oxide Semiconductor Field Effect Transistor (MOSFET) has played an important role in the prosperity of semiconductor technology. The continuous downscaling of the transistor feature size and improvements in circuit design and computer architecture have enabled the significant advances in computing. However, several future challenges in design, test, device, technology, and manufacturing have been predicted by International Technology Roadmap for Semiconductors (ITRS)[1]. The increasing leakage current, which is the current flowing through a transistor while not switching, is one of the major challenges. As technology scales down, the supply voltage must be reduced such that dynamic power can be kept at reasonable levels and power delivery can still be performed within functional requirements. But in order to prevent the negative effect incurred on performance, the threshold voltage ($V_{th}$) must be reduced at the same or higher rate such that a sufficient gate overdrive is maintained. This reduction in the threshold voltage causes an increase in the total leakage current of a chip of about 7.5 times per generation [3], which in turn can increase the static power of the device to unacceptable levels. Figure 1.1 shows the trend of process power [2]. It can be seen that the leakage power of a process is comparable to its dynamic power in current technology. Additionally, a large leakage current has negative
effects on the stability of 6T SRAM cells and the noise immunity of dynamic circuits. The impact of leakage power is worsened by a high die temperature, so better cooling techniques which are expensive are critical in order to control both active and leakage power. Thus leakage reduction is critical to the future of the semiconductor industry. The need for controlling leakage power has driven plentiful research in understanding the underlying mechanisms of leakage current and inventing leakage reduction techniques. The dominating leakage mechanisms and some of the major issues that deteriorate these leakage mechanisms are discussed in the following subsections.

![Figure 1.1. The trend of process power. The curve with diamond shows dynamic power while that with triangle shows leakage power.][2]

1.2 Sources of Leakage Current

Leakage current is the current flowing through transistors when it is not switching. Although many sources contribute to the total transistor leakage current, there are three dominant components: subthreshold leakage, gate leakage and junction leakage. Figure 1.2 illustrates these three components taking an NMOS transistor for example.
1.2.1 Subthreshold leakage

Subthreshold leakage ($I_{\text{sub}}$) is the weak inversion conduction current between source and drain in an MOS transistor occurs when gate voltage is below $V_{\text{th}}$. Subthreshold leakage grows exponentially with the inverse of threshold voltage. The inherent degradation of short channel effect of scaled devices, such as $V_{\text{th}}$ roll-off and drain-induced barrier lowering (DIBL) lowers the effective threshold voltage and deteriorates the subthreshold leakage. $V_{\text{th}}$ roll-off is the result of reduced body effect of scaled devices. Reverse bias between the source junction and the well junction of a MOS transistor widens the depletion region and increase the threshold voltage. This is known as body effect. This body effect becomes less effective in smaller device and causes $V_{\text{th}}$ roll-off. The other short channel effect, DIBL, occurs when a high voltage is applied to the drain terminal and the depletion region of the drain interacts with source terminal to lower the source potential barrier. With DIBL, the source terminal injects carriers into the channel without applying sufficient gate voltage and, as a result, effective $V_{\text{th}}$ is lowered. Taking the short channel effect into account, the overall subthreshold leakage can be modeled as [25]:

$$I_{\text{sub}} = A \times \left(1 - e^{-\frac{V_{ds}}{v_T}}\right) \times e^{\frac{V_{ds} - V_{th0} - \gamma \times V_S + \eta \times V_{ds}}{n v_T}}$$

$$A = \mu_0 C_{OX} \frac{W}{L_{\text{eff}}} (v_T)^2 e^{1.8} e^{\frac{-\Delta V_{TH}}{n v_T}}$$

(1.1)
Where $I_{\text{sub}}$ is the subthreshold leakage, $V_{th0}$ is the zero bias threshold voltage, $W$ is the transistor width, $L_{\text{eff}}$ is the transistor length, $V_{ds}$, $V_{gs}$, and $V_s$ are the voltage difference between drain and source, gate and source, and voltage at the source, respectively. $C_{OX}$ is the gate oxide capacitance, $n$ is the subthreshold swing factor, $v_T$ is the thermal voltage, $\mu_0$ is the zero bias mobility, $\gamma$ is the body effect coefficient, and $\eta$ is the DIBL coefficient. $\Delta V_{TH}$ is the term introduced to account for the transistor-to-transistor difference in threshold voltage. It should be noted that the short channel effect plays an important role in deciding the subthreshold leakage and the shorter channel length enhance both the $V_{th}$ roll-off and DIBL while higher drain voltage enhances DIBL. Figure 1.3 illustrates the relationship between the subthreshold leakage and channel length. Nevertheless, higher surface and channel doping and shallow source/drain junction depths reduce the DIBL leakage mechanism[25].

![Figure 1.3. Subthreshold leakage as a function of gate length.](image)
1.2.2 Gate leakage

Gate leakage ($I_{\text{gate}}$) comes from the direct tunneling current which is due to the tunneling of electrons (or holes) from the substrate and source/drain overlap region through the gate oxide potential barrier into the gate. Direct tunneling through the gate or Fowler-Nordheim (FN) tunneling through the oxide bands and the hot carrier injected from substrate into oxide are the major gate leakage components. Both direct tunneling and FN tunneling are functions of the electric field across the gate oxide. The gate leakage thus can be expressed as [6]:

$$I_{\text{gate}} = A \times E_{\text{ox}}^2 \times e^{B E_{\text{ox}}}$$

(1.2)

Where $I_{\text{gate}}$ is the gate leakage, $E_{\text{ox}}$ is the electric field across gate oxide and $A$ and $B$ are the parameters taking into account the related physics parameters, such as the barrier height of conduction band and mass of electrons. Detail derivations of $A$ and $B$ can be found in [6]. As Equation 1.2 suggests, gate leakage depends on the electric field across the gate oxide which is effected by the gate oxide thickness and the oxide material. Figure 1.4 shows gate leakage as a function of the oxide thickness for oxide made of SiO$_2$ and high-K dielectric materials.

1.2.3 Band-to-band tunneling current

The third major leakage component is the junction leakage which is the tunneling current between the valence band and the conduction band in the depletion region of the junctions. It is also known as band-to-band tunneling current. Advanced MOSFETs use heavily doped shallow junctions and halo doping to achieve performance. This high doping concentration raises the junction leakage.

It has been long believed that subthreshold leakage and gate leakage dominate the overall leakage current for technologies with gate length larger than 50nm. Figure 1.5 shows the above mentioned three major leakage components extrapolated from [5]. It can be seen that the gate leakage will become comparable to the subthreshold leakage while the band-to-band tunneling current is still negligible in 30nm technology generation. Accordingly, the focus of this research will be on subthreshold leakage and gate leakage.
Figure 1.4. Gate leakage as a function of gate oxide thickness. [5].

1.3 Influence of variations in process and operating conditions

There are several variations that have strong impact on leakage current. Sources of variations include the environmental variations during operation such as supply voltage and temperature variation and process-related variations such as variations in gate length, threshold voltage, and gate oxide thickness. Since leakage current depends on the voltage applied to the terminals of a transistor, leakage grows with supply voltage and the growth is super-exponential. The process variations as a percentage of their nominal values increase when technology advances. In [4], it is shown that the leakage current can vary from the target leakage current by 6.5x when considering process variations. The measurement of chips in 0.18um technology shows that 30mV variation in threshold voltage can result in 20x difference in leakage power and 30% variation in frequency [28]. Figure 1.6 illustrates the
1.4 Overview of Thesis

In this thesis, the impact of technology scaling and process variations on the circuit level leakage reduction techniques proposed in literature are evaluated. Due to the lack of leakage solutions for interconnects, new schemes for leakage-aware crossbar designs are proposed. Additionally, an architectural level leakage simulator is
Figure 1.6. Impacts from temperature and process variations on subthreshold and gate leakage.

Chapter 2 provides a comprehensive evaluation of run-time leakage reduction techniques. Several run-time leakage reduction techniques are implemented for various circuit components including datapath logic, memory components, and interconnects. Leakage reduction effectiveness, minimum idle time required to gain overall energy savings, power and delay penalty, as well as area overhead are considered when evaluating these techniques. Spice simulations are done to justify the evaluation. The impacts of technology scaling and process variations are also observed.

Chapter 3 proposes leakage-aware crossbar designs for on-chip networks. Circuit level power optimizations utilizing dual-threshold voltage process are applied to traditional crossbar to take advantage of some of the properties of on-chip networks. The proposed schemes reduce both gate and subthreshold leakage in either active or standby mode.

Chapter 4 presents an architectural level leakage power simulator to enable
the exploration of various optimizations. The presented simulator considers the
dynamic temperature feedback and process variations which are the parameters
among that have strong impacts on leakage power. This simulator is aimed to
be used as a tool for architectural leakage power optimizations. Nevertheless, the
hierarchical leakage power model embedded in the simulator is extracted from the
circuit level which can be customized to reflect the circuit designs. The results
of the simulator presented in Chapter 4 explore the impact of temperature and
process variations on leakage power of a processor.

Chapter 5 examines the influence of leakage reduction techniques on delay/leakage
uncertainties. Process variations (such as variations in transistor channel length
and transistor threshold voltage) as a percentage of their nominal values increase
when technology advances. Many of the theories behind leakage reduction tech-
niques lie in controlling these parameters which vary increasingly across technol-
ogy scaling. Thus, the interrelationship between leakage reduction mechanisms
and process-related delay and leakage uncertainty is then investigated in Chapter
5. The results provided can serve as a guideline for the leakage and uncertainty
co-optimizations.

Chapter 6, finally, summaries the results of this work and suggests future work of
interests.

1.5 Contributions of This Thesis

The contributions of the work presented in this thesis include:

- Comprehensive analysis of leakage power reductions and the related impact
  on uncertainty for circuit designers when designing different components.

- Proposed new solutions for leakage for crossbar interconnects.

- Design of an architectural level leakage power simulator that includes the
circuit level information, process variation, and dynamic temperature feed-
back.

- Analyzed impact of leakage optimizations on delay/leakage uncertainty.
Chapter 2

Analysis of Leakage Power Reduction Techniques

2.1 Introduction

Many techniques have been proposed to achieve leakage power reduction. Some require modification of the process technology, achieving leakage reduction during the fabrication or design stage. Others are based on circuit level optimization schemes that require architectural support and/or minor technology support. Some popular technology techniques include Multiple $V_{th}$ CMOS (MTCMOS), Silicon on Insulator (SOI), strained silicon, finFET, multi-gate structure, and using Hi-K dielectric materials. MTCMOS technique [7] [8] [9] assigns low threshold voltage transistors for the devices in critical paths and needs extra process steps and masks to generate multiple threshold voltage transistors. The usage of SOI, strained silicon, multi-gate structure, and finFET is intended to improve performance while using a thicker gate oxide, so that gate leakage can be controlled. Nevertheless, the higher performance allows lower $V_{dd}$ level, which yields less drain induced barrier lowering (DIBL) and therefore less subthreshold leakage. SOI [10] [11] has a significant impact on power by virtually eliminating the diffusion capacitance and allowing for steeper subthreshold slopes. In strained silicon [12], the atoms are stretched by inserting germanium atoms into the silicon lattice and thus the atoms in silicon are moved slightly farther apart. This reduces the atomic forces
that interfere with the movement of electrons through the transistors and thus the performance is improved. Multi-gate and finFET [15] [13] [14] structure increase the control over the channel and a thicker gate oxide can be used. The other process solution for leakage reduction is using Hi-K dielectric material for gate oxide to mitigate the gate leakage for the same physical oxide thickness, due to a higher oxide energy barrier [16]. Another category of leakage reduction techniques includes those technologies demanding modifications in circuit designs and applying at run-time dynamically. The run-time leakage reduction techniques are based on reducing the leakage by changing the bias conditions in the four terminals of a transistor and can be generalized into three categories: input vector control [32] [17][18], threshold voltage control [23] [24][19] [20], and power supply gating [21][22].

There is some work discussing the effectiveness of leakage reduction techniques as technology scales. In [26], a model and device measurements predicting the scaling nature of the stacking effect were presented. The decreasing effectiveness of BBC with scaling was shown in [27] using transistor and test chip leakage measurements. However, the influence of design style and some other issues brought by technology scaling have not been considered in these works. One of these issues is the sensitivity of leakage power to the process variations in gate length and threshold voltage. It has been shown that the 30mV variation in threshold leakage can result in 20x difference in leakage power in 0.18um technology [28]. The impact of process variations becomes even severe in scaled technologies and should be considered when evaluating the leakage reduction techniques. Moreover, it is expected that hi-K dielectric materials will be used in more aggressive technologies. While the maturity of hi-K dielectric materials is still under debate, the contribution of gate leakage to the total leakage remains indisputable. We focus our study on run-time leakage reduction techniques applied to different functional units in datapath, memory structures and interconnects designed using different design styles. Our goal is to examine the effectiveness of currently used leakage reduction techniques in future technologies, considering the scaling impact not only on the leakage reduction effectiveness but also on the incurred overheads. With the availability of all the functional units in datapath, memory structures and interconnects, the
analysis provides the comprehensive prediction and validation for the implications of technology scaling to the run-time leakage reduction techniques. Part of the data presented in this chapter are published in [33], [34] and [35].

2.2 Run-Time Leakage Reduction Techniques

The run-time leakage reduction techniques are based on reducing the leakage by changing the bias conditions in the four terminals of a transistor. We can generalize them into three categories:

2.2.1 By input vector control

Many researchers have used models and algorithms to estimate nominal [29] and minimum and maximum leakage of a given circuit [30]. This work has made evident the influence of the input pattern on the circuit leakage behavior, which is a consequence of the 'stacking effect' [31]. As the state of devices in the stack is determined by their corresponding inputs, which in turn are determined by the unit’s input signals, the goal can be expressed as finding the input pattern that maximizes the number of disabled transistors in all stacks across the unit. Once this vector is found, we can switch the input vector to this minimum leakage input when the unit is idle for a period of time. The implementation of the input vector control technique requires minimal architectural support. The sleep signal that determines whether the device is active or not may be already implemented in most designs but we still need to determine the threshold of idleness beyond which the input vector control is beneficial as there is an overhead energy associated with the transition to sleep (low leakage) mode.

2.2.2 By increasing the threshold voltage

This technique has different implementations, but all of them require some process technology support to change the threshold voltage of some (or all) transistors from the default defined for the technology. Some implementations in this category includes Multiple Threshold Voltage CMOS (MTCMOS), which assigns low
threshold devices in the critical path while high threshold devices are used in non-critical path, Dynamic Threshold MOS (DTMOS), in which the body and gate of each transistor are tied together such that whenever the device is off, low leakage is achieved while when the device is on, higher current drives are possible, and Variable threshold CMOS (VTCMOS), which raises $V_{TH}$ during standby mode by making the substrate voltage either higher than $V_{dd}$ (P devices) or lower than ground (N devices).

### 2.2.3 By gating the supply voltage

The last approach considered is power supply gating. There are many ways in which this technique can be implemented, but the basic idea remains: to lower or shut down the power supply so that the idle units consume less or no leakage power. This can be done by inserting "sleep transistors" to cut the path from the power supply to the units or by controlling the supply voltage regulators. The latter can also support Dynamic Voltage Scaling (DVS), which is a popular technique for dynamic power management.

### 2.3 Experimental Setup

From the techniques described in Section 2.2, one per category has been chosen for targeted designs. To obtain a comprehensive analysis of the effectiveness of each technique, the following major units were custom designed with their power characterizations listed in Table 2.1: a 32-bit adder, a 16x16 multiplier, a 32-bit shifter, a 9-bit multiplexer, various 32-bit wide Boolean logic functions, a 128-bit SRAM array and a 32-bit8mm bus. BSIM3 model is used as the device model for 0.25$\mu$m, 0.18$\mu$m, and 70nm while BSIM4 model, which includes gate leakage, for 65nm. The simulation results in 65nm technology (with gate leakage) are compared against that in 70nm technology (without gate leakage) to reflect the impact of technology improvement, especially the inclusion of hi-K dielectric materials. Note that the gate oxide material is assumed to be SiO$_2$. Due to the much higher energy required for hole tunneling in SiO$_2$, gate leakage for a PMOS device is typically one order of magnitude smaller than an NMOS device with identical thickness
of gate oxide layer. For all designs and experiments, MicroMagic MAX is used for layout creations and HSPICE for circuit-level simulations on the conditions listed in Table 2.2. The possible leakage reduction is directly estimated from SPICE simulation. Take note that Short Channel Effect (SCE) and Drain Induced Barrel Lower (DIBL) have been considered when obtaining the leakage power. The implementation of each technique is described in the following.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Average Leakage Power (nW (%of dynamic power))</th>
<th>Average Dynamic Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25</td>
<td>0.18</td>
</tr>
<tr>
<td>A</td>
<td>712 (&lt;0.01)</td>
<td>1550 (0.01)</td>
</tr>
<tr>
<td>B</td>
<td>1946 (&lt;0.01)</td>
<td>2286 (0.01)</td>
</tr>
<tr>
<td>C</td>
<td>5760 (0.07)</td>
<td>8680 (0.26)</td>
</tr>
<tr>
<td>D</td>
<td>16.3 (&lt;0.01)</td>
<td>29.4 (&lt;0.01)</td>
</tr>
<tr>
<td>E</td>
<td>14.4 (&lt;0.01)</td>
<td>28.7 (0.01)</td>
</tr>
<tr>
<td>F</td>
<td>26.9 (0.01)</td>
<td>24.9 (0.06)</td>
</tr>
<tr>
<td>G</td>
<td>31.2 (&lt;0.01)</td>
<td>30.9 (0.03)</td>
</tr>
<tr>
<td>H</td>
<td>59.1 (&lt;0.01)</td>
<td>54.1 (0.11)</td>
</tr>
<tr>
<td>I</td>
<td>74.6 (0.01)</td>
<td>92.2 (0.06)</td>
</tr>
<tr>
<td>J</td>
<td>44.1 (&lt;0.01)</td>
<td>66.16 (&lt;0.01)</td>
</tr>
<tr>
<td>K</td>
<td>368.2 (&lt;0.01)</td>
<td>479 (&lt;0.01)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Technology</th>
<th>$V_{dd}$</th>
<th>$V_{th} (n/p)$</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25um</td>
<td>2.5V</td>
<td>470mV/-590mV</td>
<td>85°C</td>
</tr>
<tr>
<td>0.18um</td>
<td>1.8V</td>
<td>445mV/-447mV</td>
<td>85°C</td>
</tr>
<tr>
<td>0.07um/0.065um</td>
<td>1.0V</td>
<td>200mV/-220mV</td>
<td>85°C</td>
</tr>
</tbody>
</table>

Table 2.2. Summary of simulation conditions.

2.3.1 Input vector control

2.3.1.1 Datapath logic

For the datapath logic circuits, the assumption is that all designs are front-ended by latches. This assumption is reasonable as most functional units are normally used in pipelined datapaths. An implementation of the input control logic with reasonable area overhead is shown in Figure 2.1. The latch hard-wired with pre-stored value shown in the left is the control to 1 logic while that in the right is the control to 0 logic. In this design, it is guaranteed that there is always stacking in the paths to $V_{dd}$ or GND when in sleep mode. This property reduces the leakage
power of the control logic by ten folds while maintain the performance of latches and thus realizes its feasibility.

In [32], 59 random input vectors were shown to achieve a 95% confidence of finding the input vector producing the least leakage current. The key to their approach was the fitting of a Gaussian distribution to the leakage profile obtained by the selected input vectors. In our approach, 180 random input vectors were generated to fit a Gaussian distribution of leakage measurement. Each input vector was simulated by HSPICE to find the input vector with the least leakage and highest obtainable savings.

![Figure 2.1. Modified low-leakage latches with optimum sleep values stored (1 left, 0 right).](image)

### 2.3.1.2 Memory structures

The major portion of leakage power in a memory structure is consumed in the memory arrays. Applying *IVC* directly to any input ports in memory structures provides negligible savings due to the symmetric structure of SRAM cells in the array. However, a technique called leakage-biased bitlines (*LBB*) [36], which mitigates the bitline leakage flowing through the access transistors, is based on a concept similar to that of *IVC*. Instead of forcing the bitlines of inactive subbanks with a sleep vector, it simply turns off the hi-$V_{th}$ NMOS precharging transistors and lets the bitlines float. The leakage current from the bit cells automatically
bias the bitlines to a mid-rail voltage that minimizes the bitline leakage current. We evaluate this technique as the IVC scheme for memory structure by delaying the precharge of bitlines in SRAM cell arrays.

2.3.1.3 Interconnects

Interconnect wires are intrinsically leakage current-free. However, to shorten the delay time, repeaters and buffers are normally inserted along the interconnect wire. Normally, the repeaters and buffers are constructed with inverter chains optimized for speed which consume large leakage current due to its lack of stacking. However, no savings in leakage can be achieved by IVC due to the symmetric structure of normal inverters. We have employed the staggered threshold voltage buffers so that there is input pattern dependency in the interconnect wires [37]. The transistors in staggered threshold buffers are selectively assigned high threshold voltage to achieve leakage reduction while maintaining reasonable performance. This method takes advantage of the trade-off between power and delay. The staggered buffers are inserted in a 32-bit 8mm bus and IVC technique is applied to the bus during sleep mode. The buffers are inserted every 100µm and sized for speed.

2.3.2 Body bias control

VTCMOS is used as the sample technique for the category in which techniques change threshold voltage to achieve leakage savings. VTCMOS requires architectural support and does not rely completely on hardware design choices and placement, allowing it to be applied at runtime. This is a required feature for useful comparison against the other techniques studied. To provide the substrate bias, we modified the netlists generated from the layouts and manually adjusted the body voltages of P and N devices, which, by default, are wired to $V_{dd}$ and ground, respectively. Study has been done to find the optimized substrate bias level to apply in the sleep mode. For the technologies which no optimized substrate bias exists, for example, 70nm and 65nm technologies, 1.4 times of $V_{dd}$ level is used for reliability’s concern. The optimized substrate bias level for each circuit in each technology is presented in Table 2.3.
<table>
<thead>
<tr>
<th>Technology</th>
<th>Substrate Bias Level (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25</td>
</tr>
<tr>
<td>32-bit Carry Lookahead Adder</td>
<td>0.5</td>
</tr>
<tr>
<td>16x16-bit Array Multiplier</td>
<td>1.0</td>
</tr>
<tr>
<td>32-bit Shifter</td>
<td>1.0</td>
</tr>
<tr>
<td>3-to-1 Multiplexer (9-bit)</td>
<td>0.8</td>
</tr>
<tr>
<td>32 2-input XOR (32-bit word)</td>
<td>1.0</td>
</tr>
<tr>
<td>32 2-input NAND (32-bit word)</td>
<td>0.5</td>
</tr>
<tr>
<td>32 2-input AND (32-bit word)</td>
<td>0.5</td>
</tr>
<tr>
<td>32 2-input NOR (32-bit word)</td>
<td>1.0</td>
</tr>
<tr>
<td>32 2-input OR (32-bit word)</td>
<td>0.5</td>
</tr>
<tr>
<td>128-bit SRAM Array</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 2.3. Optimized substrate bias level for the units evaluated.

2.3.3 Power supply gating

2.3.3.1 Datapath logic

In the approach implementing power supply gating (PSG) for datapath logics, a Phase-Locked Loop (PLL) circuit with a voltage follower as shown in Figure 2.2 is use as the voltage regulator to set the supply level to GND level in sleep mode. In this figure, two situations are possible. The sleep signal provides a way to perform global leakage reduction by shutting down the PLL and consequently all supply voltages that depend on the reference voltage generated ($V_c$), while the enable signal in the buffer provides support for local supply gating of only the units being powered by that particular buffer.

![ PLL as a voltage regulator.](image)
2.3.3.2 Memory structures

For the memory structures, a sleep transistor is inserted between the supply and cells to control the transition between active and sleep mode. The main benefit for choosing this technique is that the data can be preserved by correctly sizing the sleep transistor. Due to the regular structure of the SRAM array, the sizing of the sleep transistor can be done efficiently.

2.4 Technology Scaling Impact Analysis

In this section, the simulated/estimated values of the effectiveness and overheads in terms of power penalty, area and performance incurred by each technique are analyzed. The results shown in this section are acquired under the assumption of no process variations.

2.4.1 Input vector control

2.4.1.1 Datapath logic

It is predicted that the "stacking effect" will be more efficient for smaller technologies, which implies the improving effectiveness of IVC with technology scaling. The reason behind this is the increasing prominence of Drain Induced Barrier Lowering (DIBL). The HSPICE results in 0.25um and 0.18um technologies shown in Table 2.4 confirm this prediction. IVC can mitigate both subthreshold leakage and gate leakage. However, the best input vectors for these two leakage mechanisms are different. Thus, as can be seen from the data of 0.07um and of 0.065um technologies in Table 2.4, the savings depends on the percentages of these two leakage mechanisms as well as the design styles.

In terms of power overhead, the only contribution comes from the transition from the state in which the unit was, to the minimum-leakage state once the unit enters the sleep mode. Note that if the switching incurred in setting the input to the desired pattern causes the dynamic power consumption larger than that of the leakage at the current state for the given idle time, there will no savings. In other words, the amount of time that the unit remains idle must be long enough so that the dynamic power used in setting the low-leakage input is less than the consumed
leakage power during the same time if no low-leakage input is set. In [38], this 
minimum idle time is formulated as:

\[ t_{\text{idle}} \geq \frac{E_{tr1} + E_{tr2} + P_{\text{leak,avg}} \times (t_{tr1} + t_{tr2})}{P_{\text{leak}} - P_{\text{leak,n}}} = \frac{E_{tr} + P_{\text{leak,avg}} \times (2t_{tr})}{P_{\text{leak}} - P_{\text{leak,n}}} \]

Where all the parameters are shown in Figure The technology scaling impact 
on power overhead is evaluated by the measurements from HSPICE simulation in 
different technologies. Due to the increasing leakage reduction, the minimum idle 
time decreases with technology scaling.
The area overhead can be hidden if the unit is front-ended with latches and thus 
is small for the function units as shown in Table 2.4; otherwise, the area overhead 
should include the area of the whole circuit shown in Figure 2.1. Note that the 
area overhead is fixed across technologies.

![Image](image.png)

**Table 2.4.** Various performance parameters of IVC. The performance penalty is less 
than 1 cycle. A:32-bit Carry Lookahead Adder, B:16x16-bit Array Multiplier, C:32-bit 
2.4.1.2 Memory structures

The simulation results of applying LBB on a 32x4-bit SRAM cell array are shown in Table 2.5. Different from IVC applied on datapath logic, the efficacy decreases as technology scales. Comparing the savings achieved in 70nm with 65nm technologies, LBB performs somewhat better when considering gate leakage since the bias condition of the SRAM 6-T cell in floating state incurs less gate leakage than that in the precharge state. The transition energy penalty happens when restoring the charge back to the bitlines before the memory cells can be used and wakeup latency is the precharging time, which is delayed until the subbank needs to be accessed. Since there is no extra hardware needed to implement this technique, there is no area overhead.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Leakage Reduction (%)</th>
<th>Transition Energy (fJ)</th>
<th>Min. idle time(µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32x4 SRAM array</td>
<td>50.1</td>
<td>0.25</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 2.5. Various performance parameters of LBB on a 32x4 SRAM array.

2.4.1.3 Interconnects

The results of applying IVC on a 32-bit 8mm bus is shown in Table 2.6. The savings increases with technology scaling. However, due to the high switching penalty of switching the whole bus wire, the minimum idle time is comparable large to applying IVC on datapath logic or memory structures. The delay penalty shown is the increase in the delay of bus with staggered buffers comparing to that of bus with normal buffers. This is the result of selectively using high threshold voltage transistors. However, use of staggered buffers provides not only the static leakage power savings shown in Table 2.6 but also active leakage savings (not shown). Comparing the results in 70nm and that in 65nm, we can see that the leakage reduction is smaller when gate leakage, which is almost independent of threshold voltage, is considered. The area overhead of applying this technique is negligible since only a stacked PMOS (a stacked NMOS) hard-wired to V_{dd} (GND) is needed to control to 1(0) along the long bus.
Table 2.6. Various performance parameters of $IVC$ on a 8mm bus with staggered buffers. The performance penalty is less than 1 cycle.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Leakage Reduction (%)</th>
<th>Delay Penalty (%)</th>
<th>Min. idle time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25</td>
<td>0.18</td>
<td>0.07</td>
</tr>
<tr>
<td>32-bit 8mm bus</td>
<td>10.1</td>
<td>30.28</td>
<td>56.35</td>
</tr>
<tr>
<td></td>
<td>5.56</td>
<td>0.88</td>
<td>0.83</td>
</tr>
</tbody>
</table>

2.4.2 Body bias control

Table 2.7 shows the various performance parameters for $BBC$. Note that we apply $BBC$ on all the datapath logic and memory structure but not on interconnects. Our results confirm the decreasing effectiveness of $BBC$, which is reasoned by the reduced control ability of substrate bias over $V_{th}$ while technology scales. This is due to the increased effect of SCE and $V_{th}$ roll off. Despite of the decreasing effectiveness of $BBC$, the data in Table 5 shows the minimum idle time reduces. This is because of the larger percentage of leakage. Note that the effectiveness of $BBC$ when being applied to NAND and AND is smaller compared to that of other designs. This is the example showing the effect of implementations with different design styles. We implement NAND and AND with pass transistor logic. The raised threshold voltage increases the leakage current of ”on” NMOS’s (PMOS’s) which pass a weak 1(0) or transistors in pass-transistor logic while reduces that of ”off” transistors. However, this situation diminishes in smaller technologies. Another observation is that since threshold voltage has negligible impact on gate leakage, the efficacy of $BBC$ is expected to be even significantly less when taking gate leakage into consideration.

The power overhead is represented by the circuitry in charge of adjusting the body bias voltage. The circuit presented in [39] uses a charge pump to change the substrate level to an optimum standby bias and a charge injector to perform the recovery to active mode in reasonable time while trying to keep the area overhead to a minimum. In this implementation, there is a portion of the circuit that continuously draws current from the supply, but its effect can be ignored due to small magnitude (around 1nA and can be kept small with careful design as technology scales down). The bulk of the power overhead is in the energy required to charge the substrate when the system is entering a sleep mode. Since the transition time for fully charging the substrate is comparatively longer, there are
two cases to be considered. Independent of how fast the substrate is charged, the energy required to charge the substrate can be estimated as:

- for $t_{idle} \geq t_{sleep}$ where the substrate is fully charged to the optimum substrate bias level:

$$E_{ch\text{-}subs} = (\Delta V_{ch})^2 \times C_{sub/A} \times A$$

- for $t_{idle} < t_{sleep}$ where the substrate is partially charged to a level less than the optimum substrate bias:

$$E_{ch\text{-}subs}(t_{tr1}) = \frac{t_{tr1}}{t_{sleep}} \times (\Delta V_{ch})^2 \times C_{sub/A} \times A$$

Where $t_{tr1}$ is the period when the charge pump is charging the substrate, $t_{sleep}$ is the time for the substrate to be fully charged to the desired substrate bias level, $A$ is the area utilized and $C_{sub/A}$ is the capacitance per unit of area from the substrate to the active regions (P or N). We assume linear relationships between the transition time and transition energy and between the transition time and the obtained reduced leakage power. This assumption results in a pessimistic but safe since the deviation of the transition energy (a smaller transition energy is estimated) is less than that of the reduced leakage power (a larger reduced leakage power is estimated) and thus the estimated idle time is the worst case number. The leakage power consumed during the time the scheme is applied, $P_{avg}$, can be estimated as simply the average of $P_{leak}$ and $P_{leak,n}(t)$ for convenience. Note that since the leakage current of the substrate bias control circuitry is only 1nA, its leakage power can be neglected. The minimum idle time thus can be formulated as:

- for $t_{idle} \geq t_{sleep}$ where the substrate is fully charged to the optimum substrate bias level:

$$t_{idle} = \frac{E_{tr} - P_{leak,n} \times (t_{sleep} - t_{wakeup})}{P_{leak} - P_{leak,n}}$$

- for $t_{idle} < t_{sleep}$ where the substrate is partially charged to a level less than the optimum substrate bias:
\[ t_{idle} = - \frac{t_{sleep} \times P_{leak, J}}{P_{leak}} - \frac{2 \times E_{ch-sub}}{t_{sleep}} \]

Where \( t_{wakeup} \) is the transition time from sleep mode to active mode. The performance overhead happens when changing the substrate level. The transition time of the charging circuits can be estimated as:

\[ t_{delay} = \frac{(\Delta V_{sub} \times C_{sub})}{(W_{driving \_device} \times I_{on})} \]

Where \( \Delta V_{sub} \) is the voltage difference of substrate to be charged, \( C_{sub} \) is the substrate capacitance, \( W_{driving \_device} \) is the width of driving devices and \( I_{on} \) is the transistor saturation current. To satisfy the feasibility and to match the speed improvement of commercial products, we scale up the size of driving transistors in the charging circuit so that the delay is scaled by 0.7x per generation. The incurred area and power overhead across technologies can be estimated with the other parameters scaled using the scaling factors in [40]. The results are presented in Table 2.8. To catch the improvement in operating frequency, the area overhead increases with technology scaling.


<table>
<thead>
<tr>
<th>Technology</th>
<th>Leaking Reduction (%)</th>
<th>Transition Energy (fJ)</th>
<th>Min. idle time(µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>81.7</td>
<td>60.6</td>
<td>0.25</td>
</tr>
<tr>
<td>B</td>
<td>77.9</td>
<td>86.7</td>
<td>0.18</td>
</tr>
<tr>
<td>C</td>
<td>91.8</td>
<td>74.0</td>
<td>0.07</td>
</tr>
<tr>
<td>D</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
<tr>
<td>E</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
<tr>
<td>F</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
<tr>
<td>G</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
<tr>
<td>H</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
<tr>
<td>I</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
<tr>
<td>J</td>
<td>94.3</td>
<td>92.6</td>
<td>0.065</td>
</tr>
</tbody>
</table>

2.4.3 Power supply gating

2.4.3.1 Datapath logic

Since PSG technique with a power generator reduced the power supply level to GND level in sleep mode, the reduction of leakage power is virtually 100% cross all
Table 2.8. Performance penalty and delay time for transitions between sleep mode and active mode and area penalty for BBC.

Technologies. The power and area overhead come from the global PLL and local buffer circuitry. Since the estimation is at the granularity of the functional unit level, only the overhead of the local buffer is included giving the penalty caused by the global PLL is hidden when a whole system is considered. In contrast to what was done earlier with buffers for BBC, the driver is not sized for a constant delay overhead but to meet the corresponding unit’s average current requirements during normal operation. Due to this reason, the results in Table 2.9 show that the area overhead and buffer enable time (performance penalty) depending on the unit the scheme is applied to. However, the incurred performance and power penalty decrease with technology scaling as shown in Table 2.10. Thus the minimum idle time decreases.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Buffer Nominal Power(µW)</th>
<th>Minimum Idle Time(µS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25</td>
<td>0.18</td>
</tr>
<tr>
<td>A</td>
<td>463.96</td>
<td>437.39</td>
</tr>
<tr>
<td>B</td>
<td>3971.08</td>
<td>2955.72</td>
</tr>
<tr>
<td>C</td>
<td>186.52</td>
<td>111.05</td>
</tr>
<tr>
<td>D</td>
<td>37.24</td>
<td>32.72</td>
</tr>
<tr>
<td>E</td>
<td>32.92</td>
<td>5.05</td>
</tr>
<tr>
<td>F</td>
<td>8.68</td>
<td>2.05</td>
</tr>
<tr>
<td>G</td>
<td>18.76</td>
<td>4.39</td>
</tr>
<tr>
<td>H</td>
<td>11.80</td>
<td>3.39</td>
</tr>
<tr>
<td>I</td>
<td>23.08</td>
<td>5.72</td>
</tr>
</tbody>
</table>


2.4.3.2 Memory structures

Gated-$V_{dd}$ is used for the implementation of our PSG for memory structures. One sleep transistor is assigned to a 128-bit SRAM column. Simulation results of using PMOS, NMOS, and CMOS as sleep transistors are shown in Table 2.11 and Table 2.12. While technology scales, the effectiveness improves and the minimum idle time decreases as expected. The sleep transistor is sized to preserve the data in sleep mode, both the area and performance penalty increase for smaller technologies. The noise analysis is critical when implementing this technique due to the low $V_{dd}$ level for preserving data in cells. From the results in Table 2.11, we can see that the efficiency of Gated-$V_{dd}$ decreases when including comparable gate leakage in evaluation. Gated-$V_{dd}$ reduces both subthreshold leakage and gate leakage. However, by analyzing the bias conditions in sleep mood, we can find it to be less efficient in reducing gate leakage than reducing subthreshold leakage. The reason for the drop in efficiency when including gate leakage is due to the extra gate leakage incurred by the sleep transistors implementing Gated-$V_{dd}$. Another difference in the leakage reduction when taking gate leakage into consideration is the efficiency of different types of sleep transistor. When only subthreshold leakage is present, NMOS sleep transistor is superior than PMOS sleep transistor since NMOS can reduce the subthreshold leakage of the access transistors in a 6-T SRAM cell while PMOS cannot. However, when gate leakage is comparable to
subthreshold leakage, PMOS sleep transistor performs better since it incurs much smaller gate leakage and it reduces the gate leakage of the pass transistors while NMOS cannot. Nevertheless, PMOS sleep transistor always causes larger area overhead due to the lower mobility of holes. The implementations in [41],[42],[43], and [21] evidence that Gated-V$_{dd}$ is one of the most practical approaches to mitigating leakage power. It seems that a NMOS sleep transistor is more commonly used. However, our result indicates that the design styles and leakage mechanisms have impacts on effectiveness of Gated-V$_{dd}$ and should be considered upon making the decision of the type of sleep transistor.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Leakage Reduction(%)</th>
<th>Area Overhead(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>26.9</td>
<td>1.68</td>
</tr>
<tr>
<td>N</td>
<td>36.9</td>
<td>0.51</td>
</tr>
<tr>
<td>C</td>
<td>38.7</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Table 2.11. Power reduction and overhead of Gated-V$_{dd}$ applied to a 128-bit SRAM array.
P:PMOS, N:NMOS, C:CMOS sleep transistor.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Normalized Access Time</th>
<th>Minimum Idle Time(nS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>1</td>
<td>682.3</td>
</tr>
<tr>
<td>N</td>
<td>1.03</td>
<td>491.8</td>
</tr>
<tr>
<td>C</td>
<td>1.04</td>
<td>463.0</td>
</tr>
</tbody>
</table>

Table 2.12. Performance of Gated-V$_{dd}$ applied to a 128-bit SRAM array.
P:PMOS, N:NMOS, C:CMOS sleep transistor.

2.5 Process Variation Impact Analysis

To see the impact of the process variations on the leakage reduction schemes, Monte Carlo analysis in Hspice was performed in 65nm technology. We assumed 10% 3-sigma variations in both threshold voltage and gate length which are the dominate parameters influencing leakage current.

Results in Table 2.13 show how the process variations affect the efficiency of IVC and BBC. The parameter “efficiency impact” is defined as the savings gained
when assuming there is no process variations divided by the saving when assuming 3-sigma process variations. It can be seen that when process variations are included in evaluation, the efficiency of IVC reduces while that of BBC increases. We see this trend as follows. For IVC, when process variations are applied, the leakage gained by process-induced gate length reduction is more than the leakage eliminated with transistor stacking, for a given input vector. However, the impact depends on distribution of the variations and the resulted efficiency varies for different designs. We argue that with the inclusion of process variations, the minimum leakage vector would need to be recalculated as the leakage profile of the unit changes. For BBC, due to the exponential relation between the growth of leakage current and threshold voltage, the higher the threshold voltage, the less the increase in leakage current caused by variation in threshold voltage. BBC actually shift the mean threshold voltage higher and thus the mean leakage current is further reduced when considering process variations.

<table>
<thead>
<tr>
<th>Leakage Savings of IVC</th>
<th>Leakage Savings of BBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o pv(%)</td>
<td>w/ pv(%)</td>
</tr>
<tr>
<td>A</td>
<td>8.89</td>
</tr>
<tr>
<td>B</td>
<td>11.94</td>
</tr>
<tr>
<td>C</td>
<td>64.22</td>
</tr>
<tr>
<td>D</td>
<td>28.68</td>
</tr>
<tr>
<td>E</td>
<td>61.35</td>
</tr>
<tr>
<td>F</td>
<td>92.25</td>
</tr>
<tr>
<td>G</td>
<td>31.2</td>
</tr>
<tr>
<td>H</td>
<td>36.34</td>
</tr>
<tr>
<td>I</td>
<td>57.54</td>
</tr>
<tr>
<td>J</td>
<td>N.A.</td>
</tr>
</tbody>
</table>


2.6 Conclusions

Table 2.14 shows trends of parameters while technology scales, based on the assumption of a scaling factor of 0.7x per generation for the delay time. It should be noted that the efficiency of BBC will reduce as technology scales while that of others increase. The effectiveness decrease of BBC is due to $V_{TH}$ roll-off and elevating SCE. Note that the declining leakage reduction causes undesirable idle time needed to gain power saving. To solve this problem, larger driving devices
are recommended at the expense of the area overhead. Our results show that even though the effectiveness of BBC decreases, the reduction will be significant (50% in average for technology in which subthreshold leakage dominates and 15% in average for technology in which gate leakage is comparable to subthreshold leakage) and the minimum idle time can be tuned to a desirable value with reasonable area overhead down to 70nm technology.

<table>
<thead>
<tr>
<th>Method</th>
<th>Leakage Reduction</th>
<th>Area Overhead</th>
<th>Min. Idle Time (ratio)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Vector Control</td>
<td>Increase</td>
<td>Fixed</td>
<td>Decrease (x0.001)</td>
</tr>
<tr>
<td>Leakage-Biased Bitline</td>
<td>Decrease</td>
<td>Fixed</td>
<td>Decrease (x0.0025)</td>
</tr>
<tr>
<td>Body Bias Control</td>
<td>Decrease</td>
<td>Increase</td>
<td>Decrease (x0.58)</td>
</tr>
<tr>
<td>Supply Gating (local)</td>
<td>Increase</td>
<td>Depend</td>
<td>Decrease (x0.34)</td>
</tr>
<tr>
<td>Gated-V_{dd}</td>
<td>Increase</td>
<td>Increase</td>
<td>Decrease (x0.05)</td>
</tr>
</tbody>
</table>

Table 2.14. Comparison of impacts of technology scaling. The ratio in the Minimum idle time is the ratio of the minimum idle time in cycles in 0.18\(\mu\)m to that in 70nm. Cycle time scaled by 0.7x per generation.

The column 4 in Table 2.14 shows decreasing minimum idle time for all the techniques evaluated regardless the trends of effectiveness. This is due to the increasing percentage of the leakage power. The decreasing ratio shown is the ratio of the minimum idle time in cycles in 0.18\(\mu\)m technology to that in 70nm technology. A 0.7x (per generation) scaling factor of the cycle time is assumed. The scaling factor of minimum idle time is smaller than that of the cycle time. This promises the feasibility of these techniques even when there are less slacks of idleness resulted by the increasing operating speed.

All the evaluated techniques cause one-time delay penalty when waking up the units. However, Gated-V_{dd} is the only technique incurs run-time performance penalty due to the sleep transistor being in serious. Because of the increasing driving current requirements, even though the wake up time decreases, the simulation result shows that the run-time performance penalty increases.

The comparison of subthreshold leakage reduction techniques, when including gate leakage, suggests that gate leakage should be considered when choosing a leakage reduction scheme. Moreover, process variations are crucial to leakage power in advanced technologies and the efficiency of BBC is more consistent than that of IVC when considering process variations.
Chapter 3

Leakage-aware Crossbar Designs for Network Chips

3.1 Introduction

While Network-on-Chip (NoC) is becoming an attractive alternative to the traditional global interconnect structure to address the design challenges of future high-performance nanoscale architectures, power consumption remains a significant constraint. In the deep sub-micron era, the interconnect wires and associated driver circuits consume an increasing fraction of the energy budget of the system. Consequently, there have been several efforts in modeling and optimizing the power of on-chip interconnect. The authors in [70] develop a technique that optimizes the interconnect power through simultaneously performing floorplanning and functional unit binding and allocation. In [71], the authors evaluate several low-swing interconnect techniques and propose energy-efficient interface circuits. This work focuses on dynamic power only and the proposed driver/receiver circuits themselves consume additional leakage power. Segmented crossbars and cut-through crossbars [72] reduce the switching capacitance by reducing the effective interconnect length. However, these schemes also only focus on reducing dynamic power. Leakage power of two on-chip crossbar switches is studied in [73]; however, this work neglects the driver circuits which are proved to consume significant leakage power in a crossbar. The state depen-
dency of leakage power is exploited in [74] by forcing the unused logics in FPGA to a least leakage state. The active leakage saving relies on the fact that there are a large portion of logics in interconnect of FPGA is unused. This is not suitable for other structures.

Four widely used interconnect fabrics are studied in [67] and crossbar is found to be power efficient. Even with power-efficient crossbar, in [68], it is shown that the crossbar consumes 50% of total node power in current technologies if leakage power is neglected. As technology scales, leakage power is predicted to be comparable to switching power. In [69], the authors include the leakage model and their simulation results show that the crossbar accounts for 30% of total leakage power of routers, while buffers consume around 60%. Given that there are plenty of techniques proposed to reduce the leakage power in memory structures such as buffers and results in [69] have proven the effectiveness of applying those techniques in router buffers, leakage crossbar designs is the focus in this work.

In [37], bus encoding technique is developed so that leakage reduction is achieved through selective use of high threshold voltage transistors in the buffers (staggered threshold buffers). According to the analysis in Chapter 2, the delay penalty of bus with staggered threshold voltage buffers can be more than 10% which is prohibited from high performance systems. One can upsize the transistors to reduce the delay penalty. However, it increases both dynamic and leakage power. In this work, similar staggered threshold voltage buffers are coupled into the crossbar designs to mitigate leakage while still maintaining performance. The power issues in current existing crossbar designs are discussed followed by the proposing of circuit level power optimizations utilizing dual-threshold voltage ($V_{th}$) process which is now industrial practice. The proposed schemes reduce gate and subthreshold leakage in either active or standby mode. Part of this work is published in [75].

### 3.2 Leakage Power in NoC

Multiplexer tree crossbar and matrix crossbar are two common crossbar implementations. The matrix crossbar is composed of fewer transistors and thus consumes less leakage power. The architecture of matrix crossbar for NoC is shown in Figure 3.1. The example shown is a 5x5 crossbar with F-bits wide flit size. There are
F physical bits in each port. The switches in the crossbar can be implemented with tri-state buffers or pass transistors. Designs with pass transistors are chosen in this study due to its properties of low leakage: requiring minimum transistor count and minimum transistor sizing. The zoom-in insert in Figure 3.1 shows an implementation of the pass transistor switches. In the output driver, the feedback transistor \( P_1 \) is required to restore the \( V_{dd} \) level to prevent from short current flowing through \( I_1 \). Note that \( P_1 \) should be sized to guarantee correct functionality. The feedback transistor is typically sized the same as the pass transistor.

To analyze the leakage, subthreshold leakage and gate leakage in the pass transistors and the output driver when output is in logic 1 and when in logic 0 are measured by Spice simulations. Figure 3.2 shows the breakdown of the leakage in one port of crossbar. The dashed lines are the results when output is in logic 0. To achieve high speed, the output drivers should be sized larger while minimum size can be used in the pass transistors. The drivers are composed of large inverters and consume significant leakage due to their size and lack of stacking. This results in the drivers being the major leakage consumer in crossbar. Another observation is that the leakage in the driver is higher when output is 0. This is due to the leakage in the feedback transistor.

### 3.3 Leakage-Aware Interconnect

Multi-threshold voltage technique is one of the most attractive solutions for controlling leakage while still maintaining high performance. In this section, how the multi-threshold transistors can be assigned in traditional pass transistor crossbar designs is first discussed. A fine grain leakage optimization will also be presented followed by the propose of three schemes that mitigate leakage at the circuit level.

#### 3.3.1 Dual-\( V_{th} \) Feedback Crossbar (DFC)

As can be seen from Figure 3.3, only the feedback transistor is in non-critical path for assigning high threshold voltage. Figure 3.3 shows the dual-\( V_{th} \) feedback crossbar (DFC) switch with output driver of one output port. The notations of transistors used in this chapter are also shown. To manage the leakage power in
standby mode, a sleep transistor $N_5$ which is controlled by sleep signal is added. This sleep transistor is shared by all the bits in a flit and it incurs negligible area overhead since wires dominate the area. While a router will be idle for a given amount of idle time, the sleep signal is set to HIGH and node A is pulled to GND.
level reducing the gate leakage of the pass transistors ($N_1$-$N_4$).

For this dual-$V_{th}$ design, the inverters in the output driver consume most of the leakage power. An efficient technique to control leakage is to force the nodes in a minimum leakage state in standby mode. However, due to the symmetry of inverters $I_1$ and $I_2$ in the output driver, forcing the input of the driver to either 0 or 1 results in the same amount of leakage current in the driver. This issue is addressed by using asymmetric $V_{th}$ inverters as discussed in the following subsection.

![Schematic of output to PE direction path of DFC.](image)

**Figure 3.3.** Schematic of output to PE direction path of DFC.

### 3.3.2 Dual-$V_{th}$ Precharge Crossbar ($DPC$)

To address the above mentioned minimum leakage state problem, dual-Vt precharged crossbar ($DPC$) is proposed. The main idea of $DPC$ is to have interconnects that have smaller delay time for one polarity of data than the other. By doing so, the output driver can be designed with asymmetric $V_{th}$ transistors to favor the speed of the other polarity of data. To achieve this, the output wire is
pre-charged to a predefined state, $V_{dd}$ in this example, so that it has virtually zero delay time for data in logic 1 state. In other words, only the path for passing data in logic state 0 is the critical path in pre-charged to HIGH crossbar design. Similar pre-charging technique has been used in buses to reduce the delay time. However, instead of sizing the inverters in drivers asymmetrically (up size the PMOS or NMOS), we use asymmetric-$V_{th}$ leakage-aware inverters. The pre-charge transistor eliminates the above mentioned size requirement for level restoration. Note that during pre-charging, the pass transistors are turned off and thus the input wire is isolated from the output wire. The pre-charge transistor has only to pre-charge the output wire to the predefined level and thus can be of small size when achieving small pre-charge time. The resulting total leakage in $DPC$ (including the high $V_{th}$ shared pre-charge transistor) is smaller than that in $DFC$ which has distributed feedback transistors.

Figure 3.4 shows an example of the output to PE port of one case where a node is pre-charged to HIGH in $DPC$. The output driver has high $V_{th}$ in the PMOS of $I_1$ and in the NMOS of $I_2$. This results in a minimum leakage state which is when $A$ is in state 0. A simple implementation is to pre-charge the output wires by transistor $P_1$ in the negative phase of the clock signal eliminating the delay penalty for low to high transition. When there is no request sent to the arbiter from all the input buffers or when in sleep mode, the pre-charge signal (pre) is deactivated to prevent switching power penalties. All bits in a single output port share one pre-charge signal. As the number of bits in a flit is usually much larger than the number of output ports, the overhead of the pre-charge control circuit is limited. Taking a design with 128-bit flit for example, the extra wire used for routing the pre-charge signal to each bit is $1/128$ (0.78%). The use of the clock signal for pre-charging, as shown in Figure 3.5, incurs loading penalty to the clock signal. To estimate the lower bound of this clock routing overhead, the depth of the input and output buffers, which are the major source of clock loading in a crossbar switch, is assumed to be one. Under this assumption, the overhead is estimated to be around 0.4% of the overall loading capacitance driven by clock signal in a crossbar switch which has 128-bit flit. Figure 3.5 shows the logic of the pre-charge signal and a snapshot of the timing of the $DPC$ scheme. The timing of $DFC$ is also shown for comparison. Pre-charging the output wire to HIGH results
in an extra transition in the output wires when there are successive LOW data to pass to the output wires. From [74], it is expected that the static probability of a signal approaches either 0 or 1. The static probability is defined as the fraction of time a signal spends in the logic 1 state. For applications that have majority of data is in logic 1, this switching power penalty can be limited. Furthermore, since the use of logic 0 provides the least leakage state, the savings of leakage amortizes the switching power penalty. For applications with the majority of data in logic 0, DPC which is pre-charged to LOW shown in Figure 3.4 should be used. For applications where data probability is even, coding/decoding circuit to code the majority of data into 0 or 1 as in [37] is recommended. Note that the performance penalty happened in [37] due to the use of staggered threshold voltage buffers is removed in our scheme through pre-charging.

![Diagram](Figure 3.4. Schematic of output to PE path of DPC design. Both of pre-charged to HIGH and pre-charge to low are shown.)
3.3.3 Segmented Dual-$V_{th}$ Feedback Crossbar (SDFC)

To further control the leakage power, segmented crossbar design is utilized in the two segmented multi-threshold crossbar schemes. The segmented crossbar has been previously proposed to reduce dynamic power and delay time [72]. In this scheme, the wires are segmented so that, for each transition, the capacitive and resistive loads are smaller. Figure 3.6 shows that first scheme: the segmented dual-$V_{th}$ feedback crossbar (SDFC). As can be seen, the path 1 (bold solid line) has smaller capacitive and resistive loads as compared to path 2 (dashed line). This provides longer slack for transitions through path 1 than that through path 2. Consequently, transitions going through the paths in the shaded area in Figure 3.6 have shorter delay time than those going through the shaded area not shaded. The longer slack removes more transistors from the critical path, allowing designers to use high $V_{th}$ transistors. This scheme not only adds more high $V_{th}$ transistors but also results in higher probability that some segments of the wires can be put in standby mode.

To minimize leakage, fewer inverters in the output and segmentation drivers should be used. It should be noted that due to the low supply voltage in advanced
technologies, for example, 45nm technology, cascading two pass transistors results in unacceptable threshold voltage drop for passing a weak signal (weak 1 when passing 1 with NMOS). In the design proposed, instead of using two inverters in each driver, a level restoring circuit is inserted between two pass transistors to prevent signals going through two cascading pass transistors while minimizing the number of inverters. Note that in the left lower part of the crossbar shown in Figure 3.6, all the pass transistors are attached using a level restoring circuit to guarantee the polarity of the output data.

![Figure 3.6. Segmented dual-V\textsubscript{th} feedback crossbar design.](image)

### 3.3.4 Segmented Dual-V\textsubscript{th} Pre-Charged Crossbar (SDPC)

Another scheme is segmented dual-V\textsubscript{th} pre-charged crossbar (SDPC) design. An example of the pre-charged to high SDPC is shown in Figure 3.7. The longer slack in the paths in the shaded area allows all transistors in their output drivers to be
of high $V_{th}$. Moreover, the use of pre-charge transistors eliminates the threshold voltage drop limitation mentioned in Section 3.3.3 and thus no level restoration requirement. The pre-charge transistor and the sleep transistor are shared by all bits in a flit.

![Figure 3.7. Segmented dual-$V_{th}$ pre-charge crossbar design.](image-url)
3.4 Experiments and Results

Leakage savings as well as the penalty in dynamic power and delay time of the proposed designs are evaluated. In this section, the experimental setups are described and the results are discussed.

3.4.1 Setups

The proposed schemes for a 5-by-5 matrix crossbar design with 128 bits per flit are implemented. The layout are designed in 45nm technology and the interconnect properties, such as wire pitch, space, aspect ratio, and dielectric material parameters, are based on the ITRS roadmap [1]. The interconnect resistance and capacitance are predicted by the interconnect model of Berkeley Predictive Technology Model (BPTM) [53]. The circuit netlist with interconnect capacitive and resistive loads is simulated with BSIM4 model using SPICE. The simulation conditions are summarized in Table 3.1 while the evaluated schemes in Table 3.2. The scheme $SC$, whose circuit is the same as the $DFC$ except for using a single nominal $V_{th}$, is implemented as the base case for evaluating the proposed schemes. To predict the power consumptions for different system requirements, simulations are done in both 1GHz and 3GHz.

| $V_{dd}$ | $|V_{th_{nom}}|/V_{th_{high}}$ | Freq. | Temp | Wire pitch & space | Wire aspect ratio |
|---------|-------------------------------|-------|------|---------------------|------------------|
| 1V      | 0.2V/0.35V                    | 1GHz/3GHz | 100°C | 0.165um             | 2.7              |

Table 3.1. Summary of simulation conditions.

| $SC$ | single $V_{th}$ traditional feedback crossbar. $|V_{th}|=0.2V$. |
|------|--------------------------------------------------|
| $DFC$ | Dual $V_{th}$ feedback crossbar with traditional crossbar architecture |
| $DPC$ | Precharged dual $V_{th}$ crossbar with traditional crossbar architecture |
| $SDFC$ | Dual $V_{th}$ feedback crossbar with segmented crossbar architecture |
| $SDPC$ | Precharged dual $V_{th}$ crossbar with segmented crossbar architecture |

Table 3.2. Summary of evaluated crossbar schemes.

3.4.2 Results

In this section, simulation results of leakage savings, performance and dynamic power of each scheme are discussed. The experiment results illustrating the sensi-
tivity of the schemes on the flit size are also presented.

3.4.2.1 Power, delay and leakage savings

Simulation results are summarized in Table 3.3. Both DFC and DPC have shorter fall time and longer rise time as compared to SC. This is due to the use of high $V_{th}$ feedback/pre-charge transistors. However, as the maximum of fall and rise times decides the cycle time, both DFC and DPC incur no delay penalty. In SDFC and SDPC, the original non-critical paths turn out to be delay bottlenecks after including the high $V_{th}$ devices in their output drivers. One can size the high $V_{th}$ output drivers to trade area and power for speed. Uniformly distributed data values are assumed when measuring the power consumption. This assumption is the worst case scenario for DPC and SDPC. The power numbers in Table 3.3 show that all the proposed schemes outperform SC not only in leakage power but also in total power. The pre-charged schemes, DPC and SDPC, incur switching power overhead for extra pre-charging under some data patterns and thus have higher switching power than DFC and SDFC. However, their reduction in active leakage amortizes the power penalty. Thus DPC and SDPC consume no larger total power than SC. Due to significant standby power savings, the pre-charged schemes are suitable for systems with uneven data polarity and those which have long idle periods in the routers.

It is intuitive that the inclusion of high $V_{th}$ device saves both active and standby leakage power. Comparing to DFC and SDFC, DPC and SPDC offer significantly higher standby leakage savings which are 93.68% and 95.96%, respectively. This is because the circuits are in the resulting minimum leakage state in standby mode in DPC and SDPC. Switching to standby mode incurs a switching power penalty, however. Minimum Idle Time shown in Table 3.3 is defined as the minimum amount of time that a circuit stays in idle so that the leakage saved in standby mode is more than the switching power penalty. The minimum idle time for operating at 1GHz implies that for all the proposed schemes, energy is saved by going into sleep mode after only one idle cycle. The results show that by segmenting the crossbar, not only is dynamic power mitigated but the leakage power is further reduced by 20% and 30% in SDFC and SDPC, respectively. This is achieved by the improvement in the output driver designs. Besides, segmentation also increases
the chance for a segment to be put in the standby mode for maximum leakage savings.

<table>
<thead>
<tr>
<th></th>
<th>SC</th>
<th>DFC</th>
<th>DPC</th>
<th>SDPC</th>
<th>SDPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>High to low delay time (ps)</td>
<td>61.40</td>
<td>51.87</td>
<td>53.08</td>
<td>62.81</td>
<td>54.90</td>
</tr>
<tr>
<td>Low to High / Precharge delay time (ps)</td>
<td>54.87</td>
<td>58.17</td>
<td>61.25</td>
<td>64.28</td>
<td>62.80</td>
</tr>
<tr>
<td>Active Leakage (mW)</td>
<td>39.9</td>
<td>35.9</td>
<td>22.5</td>
<td>23.1</td>
<td>14.5</td>
</tr>
<tr>
<td>Active Leakage Savings</td>
<td>-</td>
<td>10.13%</td>
<td>43.7%</td>
<td>42.09%</td>
<td>63.57%</td>
</tr>
<tr>
<td>Standby Leakage (mW)</td>
<td>33.99</td>
<td>29.79</td>
<td>2.15</td>
<td>19.06</td>
<td>1.37</td>
</tr>
<tr>
<td>Standby Leakage Savings</td>
<td>-</td>
<td>12.36%</td>
<td>93.68%</td>
<td>43.91%</td>
<td>95.96%</td>
</tr>
<tr>
<td>Minimum Idle Time - 1GHz (cycles)</td>
<td>&lt; 1</td>
<td>&lt; 1</td>
<td>&lt; 1</td>
<td>&lt; 1</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Minimum Idle Time - 3GHz (cycles)</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Total Power - 1GHz (mW)</td>
<td>90.94</td>
<td>78.08</td>
<td>81.28</td>
<td>59.81</td>
<td>71.58</td>
</tr>
<tr>
<td>Total Power - 3GHz (mW)</td>
<td>182.81</td>
<td>154.07</td>
<td>180.45</td>
<td>122.18</td>
<td>168.55</td>
</tr>
<tr>
<td>Delay Penalty</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>4.69%</td>
<td>2.28%</td>
</tr>
</tbody>
</table>

Table 3.3. Summary of simulation results for evaluated crossbar schemes. The power consumptions for pre-charge schemes are obtained by assuming 50% static probability which is the worst case for power. The savings and penalty are the results compared against results of SC.

3.4.2.2 Impact of flit size

Flit size is a critical design factor in NoC architectures. The more bits in a flit the larger the size of crossbar which in turn causes larger wire loads. To study the impact of the flit size on the proposed schemes, delay and power of crossbar are evaluated by varying the flit size while fixing the device sizing. The graphs in Figure 3.8 show the delay and energy-delay product versus the flit size in the evaluated schemes. Figure 3.8(a) shows the maximum of fall propagation delay time ($t_{pHL}$) and rise propagation delay ($t_{pLH}$) of each scheme. This maximum delay time decides the clock rate that the design achieves. Since all the devices are optimized for delay time in the base case where flit size is 128 bits in the evaluation, the graphs of the propagation delay time of all schemes begins to rise at 128 bits. For flit size smaller than 128 bits, the delay is dominated by devices but not interconnect loads and thus the impact of flit size is not significant for all schemes. As can be seen from Figure 3.8(a), the maximum delay time of SC grows almost linearly with the flit size for flit sizes larger than 128 bits while that of DFC starts to rise for flit size larger than 512 bits. Among the un-segmented
schemes, the flit size has stronger impact on SC and DFC than on DPC. This is because the delay bottleneck in DPC is the pre-charge time and pre-charge time is less dependent on the flit size for the following reason. In pre-charged scheme, the pass transistor is turned off when pre-charging and thus the wire loads of the input wire are isolated from the output wire. That is, pre-charging transistor has only to charge the output wire. So, when the flit size increases, only the increase in output wire has effect on the pre-charge delay time. In contrast, in the feedback schemes, the pass transistors are turned on and thus both the increase of the loads in the input wires and output wire with increasing flit size reflects in the rise propagation and fall propagation delay time. For the segmented schemes, SDPC benefits from the above mentioned effect when pre-charging while SDFC has stronger driving capability from the devices used to segment the wires. Thus the wire loads have less impact on segmented schemes. The increasing impact of flit size demands the insertion of more drivers between wires and/or larger output drivers for larger flit sizes for SC and DFC, which in turn incurs larger dynamic and leakage power. On the other hand, the delay time of DPC, SDFC and SDPC is comparably insensitive to variations in flit size. Figure 3.8(b) shows the average delay time of each scheme for various flit sizes. The increasing trends of average delay time with flit size are similar across schemes since the impact of flit size on both $t_{PLH}$ and $t_{pHL}$ is captured in average delay time. The resulting energy-delay product shown in Figure 3.8(c) show that DPC, SDFC, and SDPC are promising for designs using larger flit sizes.

3.4.2.3 Impact of process variation

The proposed technique targets to mitigate leakage power in sleep mode. Its effectiveness in the presence of process variations is also studied. Due to the negligible impact of the process variations in the metal wires on leakage power, only the process variations in the devices (gate length, threshold voltage, oxide thickness) are considered. The 3-sigma variation in gate length, threshold voltage, and oxide thickness are assumed to be 15%, 10%, and 5%, respectively. Table 3.4 compares the leakage savings with and without process variations. As expected, the use of high threshold transistors results in more leakage savings in standby mode for all proposed schemes when considering process variations. However, the ef-
Figure 3.8. Maximum delay, average delay and energy-delay product versus the flit size. Note that SC fails to meet 1GHz timing requirement for 1024-bits flit size.

The effectiveness of reducing active leakage of DPC and SDPC reduces when including process variations, one possible reason for this is as follows. During active mode, the worst leakage occurs when the pre-charge transistor is turned on. In feedback schemes, feedback transistors let the wires self-biased while, in pre-charge schemes,
pre-charging transistor conducts a strong power supply level to the wires. Thus pre-charging worsens leakage flowing through transistors with low threshold voltage comparing to feedback transistors in active mode. When there are process variations, the phenomenon is further emphasized.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>SC</th>
<th>DFC</th>
<th>DPC</th>
<th>SDFC</th>
<th>SDPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Leakage Savings</td>
<td>-</td>
<td>10.13%</td>
<td>43.7%</td>
<td>42.09%</td>
<td>63.57%</td>
</tr>
<tr>
<td>Active Leakage Savings w/ PV</td>
<td>-</td>
<td>23.79%</td>
<td>25.65%</td>
<td>67.38%</td>
<td>67.37%</td>
</tr>
<tr>
<td>Standby Leakage Savings</td>
<td>-</td>
<td>12.36%</td>
<td>93.68%</td>
<td>43.91%</td>
<td>95.96%</td>
</tr>
<tr>
<td>Standby Leakage Savings w/ PV</td>
<td>-</td>
<td>25.00%</td>
<td>94.45%</td>
<td>70.00%</td>
<td>97.78%</td>
</tr>
</tbody>
</table>

Table 3.4. The comparison of leakage savings of each scheme with and without process variations. PV: process variation.

### 3.5 Conclusions

As existing interconnect designs in on-chip network draws significant leakage power, several dual-$V_{th}$ designs are proposed to reduce both active and standby leakage. The **DFC** saves 10.13% of active leakage and 12.36% of standby leakage while **DPC** saves 43.7% of active leakage and 93.68% of standby leakage at no delay penalty. Optimizing the interconnect structure by segmenting the interconnect and properly assigning the high $V_{th}$ transistors reduce the leakage further by 30% more in **SDFC**. **DPC** and **SDPC** target to systems which have major data transfers within the same polarity. It also shows that the proposed schemes are promising for a large range of flit sizes.
Chapter 4

An Architecture-Level Leakage Simulator

4.1 Introduction

Leakage power is projected to be one of the major challenges in future technology generations and thus estimation of the leakage power at an early design phase is essential for optimizing the total power. The measurement data from [5] shows that gate leakage will become comparable to subthreshold leakage in future technology generations. Moreover, the total leakage power is strongly dependent on circuit design styles and some environmental variations, such as on-chip temperature and technology process variations. To accurately estimate the full-chip leakage power, all of these must be considered. Some methods have been proposed to estimate the full-chip leakage power. In [54], an estimation flow that takes into account both power supply and temperature variation is proposed. However, it requires time-consuming SPICE simulations for each standard cell and is not practical for use in the early design phase. In [55], the simplified power model generated by curve fitting from HSPICE simulations is used, but no validation is presented. The authors in [56] discussed a power model taking into account temperature variation. Their power model takes into account temperature variation. However, gate leakage is not discussed and clock network which consumes significant dynamic and leakage power is not modeled. Additionally, none of these models account for
process variation which has a strong impact on leakage power. It is unclear if they include gate leakage and how they differentiate the temperature dependency between gate leakage and subthreshold leakage. Similarly, the linear regression model to estimate the leakage power proposed in [57] does not include gate leakage. Additionally, none of these models account for process variations. In this chapter, we presented an architecture-level leakage power simulator taking into account the temperature profile determined by dynamic feedback from the estimated power consumption, workload and process variations. We have generated hierarchical modeling to take into account the design styles. By using the hierarchical modeling, the breakdown of the gate leakage and subthreshold leakage at the architecture level can be characterized. While the introduction of hi-K gate dielectric material is still under debate, a tool that differentiates the subthreshold leakage and gate leakage current is beneficial when evaluating leakage reduction techniques targeted at different leakage components. The applications of this tool include dynamic thermal management and run-time adaptive leakage power optimization. Part of the materials presented in this chapter is published in [66].

4.2 Power Simulator Framework

The simulator, as shown in Figure 4.1, is built by enhancing the Trimaran [58]
framework to include an energy model. Trimaran is a compiler infrastructure that supports state-of-the-art research in compilation and simulation for Very Long Instruction Word (VLIW) architectures. The simulator can configure different architectures and, in our experiments, we parameterize it to match the Itanium [59] architecture. The cycle-level simulator of Trimaran is modified to capture the access patterns of different components of the architecture. The dynamic energy framework we use is activity based, i.e., we assume dynamic energy is consumed by a component only when that particular component is used while leakage energy is consumed every cycle.

The dynamic energy of the functional units, caches, table look-aside buffers (TLBs), pipeline units, the latches, the IO unit, and the register files are read from a table that stores the results from HSPICE simulations. The dynamic energy of the clock network was obtained using the model described in [60]. The leakage power is expressed as analytical models which are discussed in Section 4.4.

To provide a run-time temperature feedback to the energy model, the framework was further enhanced with an interface to Hotspot [61], a temperature estimation tool. HotSpot models the temperature of a microprocessor at the granularity of a Functional Unit Block (FUB) by making use of the duality that exists between heat flow and electricity. The tool is initialized by feeding it the floor plan, shown in Figure 4.2, to mimic the Itanium architecture.

### 4.3 Process Technology and Device Model

The device model used in this paper is based on 65nm technology Berkeley Predictive Technology Model (BPTM) [53]. To capture the technology trend and dynamic/leakage power accuracy, we modify the parameters according to [1] and [62] and use typical process and device parameters in BPTM. The major parameters in the model are as shown in Table 4.1. The channel ($N_{ch}$), source/drain($N_s$, $N_d$), and substrate ($N_{sub}$) doping concentrations are calculated from the 2D Gaussian S/D doping profile in [62] and are made to comply measurement results from Intel fabrication [5]. In [5], the fabricated 30nm physical gate length (equivalent to
Figure 4.2. Floor plan assumed based Itanium architecture.

65nm technology) transistors showed the $I_{on} - I_{off}$ performance with $I_{on} = 570 \mu A/\mu m$ for NMOS and $I_{on} = 285 \mu A/\mu m$ for PMOS and with $I_{off}$ around 100nA/\mu m. Note that, in [5], gate leakage is measured to be the same magnitude as subthreshold leakage and junction leakage component is verified to be negligible. Our HSPICE measurement results of the assumed device model fit in the same range and this device model is used as the base for accurate full chip leakage power estimation.

<table>
<thead>
<tr>
<th></th>
<th>Model Parameters</th>
<th>Current Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_{ch}$</td>
<td>$T_{ox}$</td>
</tr>
<tr>
<td>NMOS</td>
<td>1.5e18</td>
<td>15A</td>
</tr>
<tr>
<td>PMOS</td>
<td>1.5e18</td>
<td>15A</td>
</tr>
</tbody>
</table>

Table 4.1. The major parameters and current measurements of the 65nm device model used. Measurement is done at $V_{dd} = 1V$ and 100°C.
4.4 Hierarchical Modeling

The modeling has three levels of hierarchies: device level, circuit level, and architecture level. This hierarchy is illustrated in Figure 4.3. The modeling at each level is discussed in the following subsections.

![Diagram of 3-level model hierarchy]

Figure 4.3. 3-level model hierarchy.

4.4.1 Device level modeling

Two major leakage mechanisms: subthreshold leakage and gate leakage are considered in the device level model and are described below.

4.4.1.1 Subthreshold leakage

Subthreshold leakage is the weak inversion conduction current between source and drain in an MOS transistor occurs when gate voltage is below $V_{th}$. To account for the Short Channel Effect (SCE) and Drain Induced Barrier Lowering (DIBL), we model it as [63]:

$$I_{sub} = K \times I_0 \times \left(1 - e^{\frac{V_{ds}}{v_t}}\right) \times e^{\frac{V_{gs} - V_{th} - V_{off}}{n \times v_t}}$$ (4.1)

$$I_0 = \mu_0 \times \frac{W}{L} \times \sqrt{\frac{q \times \epsilon_{si} \times N_{ch}}{2 \times \Phi_s}} \times v_t^2$$

$$V_{th} = V_{th0} + \Delta V_{th} (SCE) + \Delta V_{th} (DIBL)$$
\[ \Delta V_{th} (SCE) = - \left( \frac{0.5 \times DVT0}{\cosh \left( DVT1 \times \frac{L_{eff}}{L_t} \right) - 1} \right) \times (V_{bi} - \Phi_s) \]

\[ \Delta V_{th} (DIBL) = - \left( \frac{0.5 \times DVT0}{\cosh \left( DSUB \times \frac{L_{eff}}{L_{to}} \right) - 1} \right) \times (ETA0 + ETAB \times V_{bs}) \times V_{ds} \]

Where \( I_{sub} \) is the subthreshold leakage, \( K \) is a fitting parameter, \( W \) is the transistor width, \( L \) is the transistor length, \( V_{ds}, V_{gs}, V_{bs} \) are the voltage difference between drain and source, gate and source, and subtract and source, respectively. \( V_{off} \) is the offset voltage in subthreshold region, \( n \) is the subthreshold swing factor, \( v_t \) is the thermal voltage, \( \mu \) is the mobility, \( \epsilon \) is the dielectric constant, \( \Phi_s \) is the surface potential, \( V_{bi} \) is the built-in voltage of the source/drain junctions, \( L_{eff} \) is the effective channel length, \( L_t \) is the characteristic length, \( DVT0, DVT1 \) are SCE coefficients, \( ETA0 \) and \( ETAB \) are DIBL coefficients, and \( DSUB \) is DIBL coefficient exponent. The parameters are physical parameters or from the numbers defined in BPTM. Detail of the definitions and calculations of the parameters can be found in BSIM4 manual[63].

To include the within-die process variation, the average leakage can be modeled by integrating the range of process variation multiplied by the sub-threshold leakage assuming normal distribution of all the process variation parameters. The approximation of the integration is done in [48] [64]. We have considered the variations oxide thickness, channel doping concentration, and gate length which are found to have most impact on subthreshold leakage. The subthreshold leakage with multiple process variation parameters is modeled as:

\[ I_{sub_pv} = I_{sub} \times e^{\frac{\sigma_{pv}^2}{2\lambda_{px}^2} + \frac{\sigma_{N_{ch}}^2}{2\lambda_{N_{ch}}^2} + \frac{\sigma_{l}^2}{2\lambda_{l}^2}} \]  \( (4.2) \)

Where \( \sigma \) is the standard deviation and \( \lambda \) is the relationship between process variation parameter and the subthreshold, which can be predicted by measuring the relationship between them in NMOS/PMOS by Hspice simulation.
4.4.1.2 Gate leakage

The major contributor of the gate leakage is the direct tunneling current which is due to the tunneling of electrons (or holes) from the substrate and source/drain overlap region through the gate oxide potential barrier into the gate. The tunneling current can be expressed as [63]:

\[ I_{gc} = W_{eff} \times L_{eff} \times J_{gate} \]  

(4.3)

\[ J_{gd} = J_{gs} = W_{eff} \times J_{gate} \]

\[ J_{gate} = A_g \times \left( \frac{t_{oxref}}{t_{ox}} \right)^{n_{t_{ox}}} \times \left( \frac{V_g \times V_{aux}}{t_{ox}^2} \right) \times e^{(-W \times t_{ox} \times (\alpha - \beta |V_{ox}|)(1+\gamma|V_{aux}|))} \]

where \( I_{gc}, I_{gd}, \) and \( I_{gs} \) are the gate to substrate current, gate to drain overlap current density and gate to source overlap current density, respectively. \( t_{ox} \) is the gate thickness. \( V_g \) is the voltage difference between gate and the terminal interested (substrate, drain, or source). \( A_g, t_{oxref}, V_{aux}, V_{ox}, A, B, \alpha, \beta, \) and \( \gamma \) are some physical factors and the detail derivations can be found in the BSIM4 model manual [63]. Similar to subthreshold leakage, the process variation impacts can be predicted by:

\[ I_{gate,pv} = I_{gate} \times e^{\frac{-V_{aux}^2}{2 \times t_{ox}^2}} \]  

(4.4)

Note that only oxide thickness is considered since it is the only parameter that has significant impact on the gate leakage [48].

4.4.1.3 Validation

The device-level model is validated through comparing it with Hspice simulation results over the temperature from 25°C to 150°C. The comparison is shown in Figure 4.4. The average error is 1.56% and 12% in average for temperature and process variations, respectively.
4.4.2 Circuit level modeling

One of the goals of these analytical models is to differentiate the subthreshold leakage from gate leakage. Analyzing the circuit of the major components in a processor and then representing them with the leakage current components shown in Figure 4.5. The circuit-level models include the breakdown of subthreshold and gate leakage. Note that we assume SiO$_2$ is used as the gate oxide material and thus the gate leakage of PMOS is negligible. The modeling is discussed in the following subsections.

4.4.2.1 Modeling for memory, latches, and clock network

The major part of clock drivers, pipeline control latches, caches and memory, register files are mainly composed of some unit circuit blocks. For the regular structure like this, the analysis is done by analyzing the bias conditions at each node for every possible input combination and taking average of each leakage current assuming each input has same probability. From the leakage power breakdown, we can see that the majority of the leakage power is mainly from clock drivers, L1 and L2 cache and pipeline control latches distributed in datapath logic. The circuit styles of these circuit blocks are regular and thus can be easily obtained.
Figure 4.5. Leakage components at different bias conditions.

The example of analyzing an inverter is shown in Table 4.2. The analysis results of other components are shown in Figure 4.6.

<table>
<thead>
<tr>
<th>Bias Condition</th>
<th>Symbol</th>
<th>Leakage (nA/um)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$I_{subn}$</td>
<td>Hi-Vth: 0.071</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nominal Vth: 75.2</td>
</tr>
<tr>
<td>0</td>
<td>$I_{gaten1}$</td>
<td>57.2</td>
</tr>
<tr>
<td>0</td>
<td>$I_{gaten2}$</td>
<td>167.4</td>
</tr>
<tr>
<td>1</td>
<td>$I_{gaten3}$</td>
<td>476.5</td>
</tr>
<tr>
<td>1</td>
<td>$I_{subp}$</td>
<td>Hi-Vth: 0.112</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nominal Vth: 111</td>
</tr>
</tbody>
</table>

Table 4.2. Circuit-level leakage analysis of an inverter.

4.4.2.2 Modeling for arithmetic circuit and random logic

Due to the irregularity of arithmetic circuit and random logic, instead of circuit level analysis, we estimate their leakage power according to their equivalent transistor width. The detail modeling is described in Section 4.4.3.2.

4.4.2.3 Validation

Each component is designed and simulated with Hspice by measuring the leakage power both the power supply and all input pins. The comparison of model and simulation is shown in Figure 4.7. The circuit-level modeling achieves 5.31% error in average in the range of 25°C to 150°C compared to Hspice simulation results.
<table>
<thead>
<tr>
<th>Schematics</th>
<th>Average Leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock driver cell</td>
<td>$I_{\text{clkdrv}} = 1/2 * (I_{\text{gates1}} + I_{\text{gates2}} + 2I_{\text{subn}} + I_{\text{subp}})$</td>
</tr>
<tr>
<td>6T L1/L2 Cache cell</td>
<td>$I_{\text{6TCell}} = 2I_{\text{gates1}} + I_{\text{gates2}} + I_{\text{gates3}} + 2I_{\text{subn}} + I_{\text{subp}}$</td>
</tr>
<tr>
<td>TLB CAM cell</td>
<td>$I_{\text{cam}} = 2I_{\text{gates1}} + 5I_{\text{gates2}} + 2I_{\text{subn}} + I_{\text{subp}}$</td>
</tr>
<tr>
<td>Pipeline latch</td>
<td>$I_{\text{DFF}} = 4.5I_{\text{gates1}} + 4.8I_{\text{gates3}} + 3I_{\text{subn}} + 3.3I_{\text{subp}}$</td>
</tr>
</tbody>
</table>

**Figure 4.6.** Leakage analysis of each structure component.

### 4.4.3 Architecture level modeling

The simulator is configured to model architecture similar to that of Intel Itanium and the circuit area and transistor counts are modeled from [59]. The architecture-level modeling is discussed below.
Figure 4.7. Leakage power of components been analyzed at circuit level. Both model and Hspice simulation results are shown.

4.4.3.1 Power modeling for memory components

Since the major power consumption of on-chip memory comes from the cell array, in our estimation, the cell in each cache level is designed and analyzed as in Section 4.4.2. The total leakage power is estimated as the product of the number of cells and the leakage power of each cell. Take L2 cache for example, the total leakage power is:

\[ I_{\text{leak}, \text{power}_{\text{L2cache}}} = N_{\text{cell}} \times V_{dd} \times I_{\text{leak}, \text{6Tcell}} \]  

(4.5)

\[ = N_{\text{cell}} \times V_{dd} \times (2 \times I_{\text{gaten1}} + I_{\text{gaten2}} + I_{\text{gaten3}} + 2 \times I_{\text{subn}} + I_{\text{subp}}) \]

Where \( N_{\text{cell}} \) is the total number of bits in L2 cache. To capture the design style impact, each cell is designed according to the design considerations below.

i  L1 I-Cache and D-Cache

To achieve 1-cycle latency for L1 cache, single-ended full-swing SRAM cell is used. Special care of the large pass transistor (for full swing of bit-line) is
needed to trade off the area and power consumption for speed.

ii L2 Cache
Standard 6-T SRAM cell is used for the L2 cache to trade the speed for area. High threshold transistors (\(|V_{th}| = 350\text{mV}\)) are used and the cell is sized to minimize the leakage consumption.

iii L3 Cache
The cell for L3 cache is similar to L2 cache with the exception of the sizing. Use of Hi-threshold voltage transistor is necessary due to the huge size of L3 cache. For the same reason, smaller sizing is used for area and power’s concern. Besides the circuit level optimization for power, we implement cache decay [65] in L3 cache to turn off unused lines.

iv TLB Cache
The design of content addressable memory (CAM) cell array is also implemented for the estimation of TLB power. The CAM cell is sized for performance.

4.4.3.2 Power modeling for memory components

I Clock network
The clock model proposed in [60] is embedded into our power simulator to estimate the power in the clock distribution network. Three-level H-tree clock distribution style as in Itanium processor [59] is assumed. The driving and wiring capacitance in each level can be expressed as:

\[
U_{avg} = \sum_{i=1}^{N-1} u^i \quad \text{and} \quad N = \frac{\ln(x)}{\ln(u)} \quad \text{and} \quad x = \frac{C_{load}}{C_{in}} \quad (4.6)
\]

Where \(U_{avg}\) is the required number of inverters, \(u\) is the minimum inverter size, \(C_{load}\) is the capacitive load that each buffer drives, and \(C_{in}\) is the capacitance of the minimum size inverter. This \(C_{load}\) can be obtained by dividing the total capacitive loading of the whole chip to the total number of the drivers. The leakage power of the clock network mainly comes from the single-stacked inverters in the buffers and can be approximated as:
\[ P_{\text{leak, clock driver}} = U_{\text{avg}} \times V_{dd} \times I_{\text{leak, inv}} \]
\[ = U_{\text{avg}} \times V_{dd} \times \frac{I_{\text{gaten}3} + I_{\text{subp}} + I_{\text{gaten}1} + I_{\text{subn}}}{2} \quad (4.7) \]

Note that the thermal profiling tool, HotSpot, used does not include the clock distribution network which consumes noticeable power. To compensate for this, we have included the power of clock network and distributed its power consumption into each functional/memory block according to their capacitive load (area) before feeding the run-time power numbers into HotSpot.

II Pipeline control latches

Pseudo-static \( C^2 \text{MOS} \) flip-flop style is used as the pipeline control latches due to its low leakage power consumption. The dynamic and leakage power is estimated as:

\[ I_{\text{leak, pipeline}} = P_{\text{leak, latch}} \times N_{\text{latch}} \quad (4.8) \]

and

\[ N_{\text{latch}} = N_{\text{insts}} \times N_{\text{ops}} \times N_{\text{stage}} \times N_{\text{opds}} \times W_{\text{bit}} \quad (4.9) \]

Where \( P_{\text{leak, latch}} \) is the leakage power of a single latch, \( N_{\text{latch}} \) is the number of the latches, \( N_{\text{insts}} \) is the number of instruction can be executed in parallel, \( N_{\text{ops}} \) is the maximum number of the operations per instruction, \( N_{\text{stage}} \) is the pipeline stages, \( N_{\text{opds}} \) is the number of operands per operation and \( W_{\text{bit}} \) is the bit width of each operand plus the number of control and result bits.

III Arithmetic and control logic

The leakage power of arithmetic circuit and random logic is comparably small. For simplicity, we make two assumptions to simplify the modeling. The first one is, same as in [48], we assume half of transistors on a chip are off while the other half are on and the second is assuming stack effect reduces the leakage power by 20% which is conservative. Thus the leakage power can be expressed as:
\[ P_{\text{leak,arth}} = V_{dd} \times (W_{eqn} \times (F_s \times I_{subn} + F_g \times I_{gaten,avg}) + W_{eqp} \times F_s \times I_{subp}) \] 

(4.10)

Where \( V_{dd} \) is the power supply, \( W_{eqn/p} \) are equivalent width of the specific circuit block which can be predicted by the transistor count times the on/off ratio of the transistors, \( F_s \) and \( F_g \) are the leakage reduction by stack effect for subthreshold leakage and gate leakage, respectively, and \( I_{gaten,avg} \) is the average gate leakage. \( F_s \) and \( F_g \) depend on the design style and can be derived by curve fitting from simulation results.

All the components in the integer unit and floating point unit, including an adder, a shifter, a multiplier, a divider, and Boolean logics, are handcrafted to get approximated transistor count and area. The leakage power can then be estimated according to Equation 4.10. We estimate the power of the pipeline logic, the branch units, and the bus logic by scaling the power numbers of the integer units according to their area ratio. The area ratio we use is from Itanium chip layout.

### 4.4.3.3 Validation

Since designs of cache and arithmetic circuits are available, we can validate the architecture-level modeling by comparing the model result with the HSPICE simulations. The comparison is shown in Figure 4.8. The average error for all the components is 6.53%.

We also performed validation for the dynamic energy models used in our design by comparing the simulator output with the published peak power values of the Itanium in 0.18um technology. The peak power consumption of 98W returned by our simulator is within 5% of the published result excluding the units that we do not model, such as the IA32 extension module, multimedia unit, I/O, and package.
Figure 4.8. Comparison between the results from model and HSPICE simulations at microarchitecture level.

4.5 Experimental Results

In this section, we present the experimental results for the power simulator. Table 4.3 summarizes the architecture configurations and conditions we assume and Table 4.4 shows the benchmarks used in the experiments. In the following subsections, the data presented are in 65nm technology which is a better technology node for examining the leakage power.

4.5.1 Power breakdown and temperature effect

Figure 4.9 shows the leakage power breakdown for the benchmarks we used. It can be seen that the leakage distribution of 132.ijpeg is different from that of other benchmarks. This is because the larger size of 132.ijpeg results in larger temperature difference between components, which in turn allows the temperature effect to reflect on the leakage power consumption. This points out the importance of including the temperature variation for leakage power estimation. Another observation is the larger leakage consumption of L3 cache when executing tomcatv. This is because there is more L3 accesses in tomcatv as compared to the other appli-
Table 4.3. Architecture configurations assumed.

<table>
<thead>
<tr>
<th>Processor Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose Static Register File</td>
</tr>
<tr>
<td>General Purpose Rotating Register File</td>
</tr>
<tr>
<td>Floating Point Register File</td>
</tr>
<tr>
<td>Floating Point Rotating Register File</td>
</tr>
<tr>
<td>Prediction Register File</td>
</tr>
<tr>
<td>Prediction Rotating Register File</td>
</tr>
<tr>
<td>Control Register File</td>
</tr>
<tr>
<td>Control Rotating Register File</td>
</tr>
<tr>
<td>Branch Target Register File</td>
</tr>
<tr>
<td>Integer Units</td>
</tr>
<tr>
<td>Floating Point Units</td>
</tr>
<tr>
<td>Branch Units</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
</tr>
<tr>
<td>L1 I- &amp; D-Cache Data Array</td>
</tr>
<tr>
<td>L1 I- &amp; D-Cache Tag Array</td>
</tr>
<tr>
<td>L2 Cache Data Array</td>
</tr>
<tr>
<td>L2 Cache Tag Array</td>
</tr>
<tr>
<td>L3 Cache Data Array</td>
</tr>
<tr>
<td>L3 Cache Tag Array</td>
</tr>
<tr>
<td>L1 TLB</td>
</tr>
<tr>
<td>L2 TLB</td>
</tr>
<tr>
<td>L3 TLB</td>
</tr>
</tbody>
</table>

Table 4.4. Benchmarks used in experiments.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Source</th>
<th># of Instruction Executed</th>
</tr>
</thead>
<tbody>
<tr>
<td>132.ijpeg</td>
<td>SPECint95</td>
<td>1694618598</td>
</tr>
<tr>
<td>tomcatv</td>
<td>SPECfp95</td>
<td>179818173</td>
</tr>
<tr>
<td>paraffins</td>
<td>Trimaran</td>
<td>177828</td>
</tr>
<tr>
<td>129.compress</td>
<td>SPECint95</td>
<td>38692124</td>
</tr>
<tr>
<td>convolution</td>
<td>DSPSTONE</td>
<td>16000065</td>
</tr>
</tbody>
</table>

The architectural cache decay technique to reduce leakage is relatively less effective.

Figure 4.10 shows the leakage breakdown when running 132.ijpeg. The steady state temperature in Kelvin is also shown. Note that some components are op-
erating at higher temperature than the others. As a consequence, components of occupying a small portion of transistor budget can consume comparable power to that of larger portions. For example, the pipeline control logic is smaller than L2 cache, but due to the 13°C temperature difference and the low threshold transistors used in L2 cache, the leakage power consumed in pipeline control logic is larger than that in L2 cache. Based on applications, the temperature difference varies.

![Figure 4.9. Leakage breakdown of benchmarks.](image)

4.5.2 Impact of process variations

In Figure 4.11, the power breakdown without process variations and that with process variations are shown. The $3\sigma$ variation is set to be 12.5%, 10%, and 5% for gate length ($L$), gate oxide thickness ($T_{ox}$) and threshold voltage ($V_{th}$), respectively. To include process variations in estimating the leakage power is important since it increases 20% 30% across benchmarks. Without considering process variations, the leakage power is underestimated.
**Figure 4.10.** Dynamic/Leakage power breakdown with temperature feedback.

**Figure 4.11.** Leakage power breakdown without process variations (left), and that with process variations (right). Result of benchmark: tomcatv.

### 4.5.3 Subthreshold and gate leakage breakdown

The breakdown of subthreshold leakage and gate leakage depends on the circuit design styles. Figure 4.12 shows the breakdown of the major components with the circuit style we assumed. The results presented include the breakdown when using dynamic temperature feedback, a fixed temperature at 100°C, and room temperature to show how the temperature can affect the breakdown of leakage mechanisms. Due to the high threshold voltage transistors used in the L2 and L3 caches, their subthreshold leakage is almost negligible. While there are many
techniques to reduce the subthreshold leakage proposed for caches, techniques for controlling gate leakage in caches are needed. One can obtain this breakdown by doing circuit-level analysis discussed in Section 4.4.2 to custom the circuit-level model for the design styles in need.

![Subthreshold and gate leakage breakdown of major components. (Benchmark:132.jpg)](image)

**Figure 4.12.** Subthreshold and gate leakage breakdown of major components. (Benchmark:132.jpg)

### 4.6 Conclusions

An architecture-level leakage power simulator based on VLIW Trimaran tool-set has been built. A temperature estimation tool is integrated into the power simulator so that the leakage power can be reevaluated at run-time using temperature feedback. With hierarchical model at the architecture, circuit, device level, the information of run-time profiling, design styles, environmental variations, and technology trend can are included. Moreover, the breakdown of gate leakage and subthreshold can be obtained and this is beneficial for evaluating leakage reduction techniques.

Our experimental results show that the leakage power will exceed 50% of the chip
power budget for most of the benchmarks. Without using any leakage control techniques, the majority of leakage power comes from the cache architecture. However, the use of high threshold transistors for cache cells and taking the temperature profiling into account, makes the other processor units with higher activity no less important from a leakage power perspective.

The process variation sensitivity of the leakage power has been stressed recently. Our simulation result shows that the leakage power increases by 20% 30% at architecture-level when accounting for nominal process variations in gate length, oxide thickness, and channel doping concentration.
Chapter 5

Influence of Leakage Reduction Techniques on Delay/Leakage Uncertainty

5.1 Introduction

The process variations (such as transistor channel length and transistor threshold voltage) as a percentage of their nominal values increase when technology advances. The increasing impact of process variations is predicted to be one of the limits for future technology scaling [1]. In [4], it is shown that the leakage current can vary from the target leakage current by 6.5x when considering process variations. The measurement of chips in 0.18um technology shows that 30mV variation in threshold voltage can result in 20x difference in leakage power and 30% variation in frequency [28]. The above mentioned works prove that the performance and power consumption of actual silicon may significantly deviate from the targeted design specifications due to variations. After manufacturing, for a chip to be accepted, it must meet a minimum frequency, and at the same time, the accepted die must meet the maximum power consumption requirement. Any die that exceeds the maximum leakage power must either be binned to operate at lower frequency or discarded. The delay/leakage uncertainties due to process variations can worsen the binning distribution and cause yield loss and reliability issues. Moreover, the
rising leakage and leakage uncertainty have made the Iddq test a challenge.

Figure 5.1. Leakage power distribution for 65nm and 45nm technologies.

Our Hspice simulation (1000 runs of Monte Carlo analysis) results shown in Figure 5.1 predict the increasing uncertainty in leakage while the leakage grows as technology advances. In Figure 5.1, the random distribution of leakage is quantified by the parameter "uncertainty" which we define as the standard deviation of delay (leakage) divided by its mean value (S.D./mean). There is a need to control the uncertainty. Many active leakage power reduction techniques are based on adjusting transistor physical parameters or reducing the effect caused by these physical parameters (such as Short Channel Effect (SCE) and Drain-Induced Barrier Lowering (DIBL)). Their impact on the delay and power uncertainties should be considered when optimizing yield, performance, and power. Moreover, process-invariant design styles will be necessary for future technologies. Some works have shown the effectiveness of leakage reduction techniques in the presence of process variations [4] [44]. The impact of process variations on the effectiveness of stack forcing is presented in [4]. In [45], the effectiveness of power and variability reduction of adaptive body bias and adaptive supply voltage is compared. The authors evaluated the techniques that adjust the body bias and supply voltage as post-silicon tuning to compensate intra-die variations. However, there is no study
that compares different leakage reduction techniques with respect to their ability to control the leakage uncertainty. In [46], the authors discussed the impacts of the dynamic power management techniques on the delay variations and fabrication yield. Nevertheless, leakage power and leakage power reduction techniques are not considered in their work. In this chapter, we examine the mechanisms behind the mainstream active leakage reduction techniques that are used in design phase and evaluate the yield, delay and leakage power uncertainties under the presence of process variations when using these mechanisms. Active leakage reduction techniques are aimed to reduce the leakage current drawn when part of the circuit is operating and thus the reduction in leakage comes with the influence in the performance of the applied circuit. As both leakage reduction techniques and statistic timing analysis emerge as standard for smaller technologies, we examine how the leakage reduction techniques impact the statistic timing analysis and suggest how these emerging techniques can be implemented simultaneously. The influence of technology scaling and temperature sensitivity is also studied. Part of the data presented in this chapter is published in [52].

5.2 Process Variation Sources and Hints for Reducing the Impacts

The electrical performance of an integrated circuit is impacted by two distinct sources of variation: environmental factors and physical factors [47]. In this thesis, we focus on the physical process variations. Among the variations in transistor parameters, variations in gate length and threshold voltage are found to have most significant impacts on circuit performance and power consumption [48][45]. For devices at technologies below 100nm, the variations in threshold voltage is caused by the doping fluctuation and thus influenced by the transistor geometry. This indicates that the effective variations in the threshold voltage is dependent on variation in gate length and can be explained by the following effects:

- **First order effect:** In [49], the standard deviation of the intrinsic threshold voltage for long channel devices is analytically modeled as:
\[ \sigma V_{th} = \left( \frac{q}{C_{ox}} \right) \times \sqrt{\frac{N_{eff} \times W_{dep}}{3 \times W \times L}} \]  

(5.1)

Where \( q \) is the charge, \( C_{ox} \) is the gate oxide capacitance, \( N_{EFF} \) is the weighted doping concentration, \( W_{DEP} \) is the channel depletion width, and \( W \) and \( L \) are the channel width and gate length, respectively. This equation is derived for long channel devices which do not exhibit short channel effects. As shown in equation 5.1, the variation in threshold is caused by the doping un-uniformity and is proportional to the square root of doping concentration and inversely proportional to the square root of device gate length and channel width. From circuit design standpoint, the variations can be reduced by increasing the channel width or gate length of devices. However, the effectiveness is limited due to the weak dependence as expressed in equation 5.1.

- **Second order effect**: The second order effect that causes the variations in transistor parameters is the result of threshold roll-off and DIBL. In nanometer technologies, the shorter device gate length reduces the effective threshold voltage to:

\[ V_{th} = V_{th0} - \Delta V_{th}(V_{th \cdot roll \cdot off}) - V_{th}(DIBL) \]  

(5.2)

Where the \( V_{th0} \) is the intrinsic threshold voltage. \( V_{th}(V_{th \cdot roll \cdot off}) \) and \( V_{th}(DIBL) \) are the drop in threshold voltage due to short channel effect and DIBL, respectively. The relation between the gate length and threshold roll-off and DIBL is exponential and thus when there are variations in gate length, the net effect is increased variation in threshold voltage. Due to the exponential dependence, by increasing the gate length, the variations in both gate length and effective threshold voltage can be efficiently controlled.
5.3 Review of Mechanisms for Reducing Active Leakage

In this section, we review the mechanisms behind mainstream active leakage reduction techniques whose influence on delay/leakage uncertainty we will expect in this work. The techniques considered include increasing gate length [50], \( V_{dd}/V_{th} \) optimization [23], body biasing [44] [45], and stack forcing [51].

- **Increasing Gate Length** From equation 5.1, increasing the gate length not only reduces the leakage power but also reduces the leakage and delay uncertainties. Besides controlling the first order effect of variations, longer gate length also reduces the second order effect by lowering threshold roll-off and DIBL.

- **\( V_{dd}/V_{th} \) Optimization** Due to the strong dependence of both dynamic and leakage power on power supply voltage, lowering supply voltage is used in most low power designs. Meanwhile, as multi-threshold processes are increasingly common, power optimization through tuning supply voltage to threshold voltage ratio (\( V_{dd}/V_{th} \)) is used to achieve power-delay trade-off.

- **Body Biasing** The body effect causes threshold voltage roll-off and in turn higher leakage power. By reverse biasing the substrate of a transistor in sleep mode, the leakage current can be reduced. For post-silicon optimization, body bias is used to tune the threshold voltage back to target value.

- **Stack Forcing** The idea of "Stack Forcing" is to break a single transistor into a stacked transistor pair and thus the DIBL of the stacked transistor pair is reduced which in turn mitigates the leakage.

5.4 Experimental Setup

To evaluate the impact of the leakage power reduction techniques on the uncertainty under the presence of process variations, Monte-Carlo Hspice simulations are done in 65nm and 45nm technologies. A branched inverter chain shown in Figure 5.2 is used as the target circuit in our exploration. This circuit is modified
from canonical test inverter chain and sized to optimize the delay and emulate critical and non-critical paths. The simulation conditions are summarized in Table 5.1. The technology files used are 65nm and 45nm Berkeley Predictive Technology Model (BPTM) [53] where subthreshold leakage and gate leakage are captured. Note that the 3-σ variation of gate length and intrinsic threshold voltage is set to be 10% and 1000 Monte-Carlo runs are simulated for each technique. The setup for each mechanism is discussed in the following.

**Figure 5.2.** A branched inverter chain as the test circuit.

<table>
<thead>
<tr>
<th></th>
<th>65nm</th>
<th>45nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{\text{gate}}$(nm)</td>
<td>65</td>
<td>45</td>
</tr>
<tr>
<td>$</td>
<td>V_{\text{th}}</td>
<td>$(V)</td>
</tr>
<tr>
<td>$V_{\text{dd}}$(V)</td>
<td>1.0</td>
<td>0.7</td>
</tr>
<tr>
<td>Temp.(°C)</td>
<td>85</td>
<td>85</td>
</tr>
</tbody>
</table>

**Table 5.1.** Simulation conditions.

### 5.4.1 Gate length biasing

The optimized gate length is the trade-off between the power, delay and area. Simulation results in Figure 5.3 show the achievable leakage power savings and leakage/delay uncertainties. In our experiment, we increase the gate length by 10% as point A shown in Fig. 3. It can be seen that increasing the gate length by 10% of minimum gate length achieves 85%, 55%, and 30% in the leakage savings, leakage uncertainty reduction and delay uncertainty reduction, respectively.
5.4.2 $V_{dd}/V_{th}$ optimization

To find the optimized operating point of $V_{dd}/V_{th}$ tuning, the design space is explored and the results are shown in Figure 5.4 to Figure 5.8. In these figures, the trends of dynamic power, leakage power, leakage power uncertainty, and delay uncertainty at various $V_{dd}$ and $V_{th}$ levels are shown. These plots show that when the power supply voltage is lower, the delay and leakage uncertainties are larger. Another observation is that the delay uncertainty is smaller with higher $V_{dd}/V_{th}$ ratio while leakage uncertainty is larger at this condition. These result in different optimized point for leakage power, delay uncertainty, and leakage uncertainty. This implies that, in nanometer technologies where both leakage power and uncertainty present major challenges, the optimal $V_{dd}/V_{th}$ ratio to use can change. Thus when selecting $V_{dd}/V_{th}$, special care is needed for the variation analysis. In our experiment, we select $V_{dd}/V_{th} = 0.7V/0.2V$ and $V_{dd}/V_{th} = 0.5V/0.16V$ for 65nm and 45nm technologies, respectively. The decision is made to minimize the leakage power so that the influence of leakage optimization on the uncertainty can be evaluated.

5.4.3 Body biasing

To employ this mechanism to reduce active leakage at design time, adaptive body bias is proposed. In this design, forward bias is applied to circuit in active mode
Figure 5.4. $V_{dd}/V_{th}$ optimization for dynamic power. $V_{dd}$ is from 0.7V to 1.2V and $V_{th}$ at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case $V_{dd}/V_{th}$=1.0V/0.2V.

Figure 5.5. $V_{dd}/V_{th}$ optimization for leakage power. $V_{dd}$ is from 0.7V to 1.2V and $V_{th}$ at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case $V_{dd}/V_{th}$=1.0V/0.2V.
Figure 5.6. $V_{dd}/V_{th}$ optimization for leakage power uncertainty. $V_{dd}$ is from 0.7V to 1.2V and $V_{th}$ at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case $V_{dd}/V_{th}$=1.0V/0.2V.

Figure 5.7. $V_{dd}/V_{th}$ optimization for delay. $V_{dd}$ is from 0.7V to 1.2V and $V_{th}$ at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case $V_{dd}/V_{th}$=1.0V/0.2V.
Figure 5.8. \( V_{dd}/V_{th} \) optimization for delay uncertainty. \( V_{dd} \) is from 0.7V to 1.2V and \( V_{th} \) at S1, S2, S3, and S4 are 0.14V, 0.16V, 0.18V, and 0.20V, respectively. The numbers in z-axis are normalized to the case \( V_{dd}/V_{th}=1.0V/0.2V \).

(for high speed) while reverse bias is used in sleep mode (for low leakage power). With forward biasing to set to low threshold voltage in active mode, the default gate length can be made longer and thus causes smaller threshold voltage roll-off and DIBL. The impact of uncertainty is similar to that of increasing gate length discussed above, and consequently not presented in more detail.

Except the mechanisms discussed above, we also evaluated “Stacking Forcing” given its significant effectiveness. Techniques combining multiple mechanisms are also investigated as optimization through simultaneously tuning multiple design parameters (i.e. tuning threshold voltage, supply voltage, stack forcing, and gate length assigning) are popular. Table 5.2 summarizes the techniques evaluated. Note that due to the large delay penalty of stack forcing, we evaluate two techniques: one that applies stack forcing to all elements and another that applies it to only the non-critical path.
Table 5.2. Leakage reduction techniques evaluated.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig</td>
<td>Original design without any optimization.</td>
</tr>
<tr>
<td>SF</td>
<td>Stack forcing to both PMOS and NMOS.</td>
</tr>
<tr>
<td>LF</td>
<td>Gate length biasing by increasing gate length by 10% of the minimum gate length.</td>
</tr>
<tr>
<td>VOpt</td>
<td>Tuning $V_{dd}/V_{th}$ to optimize leakage power.</td>
</tr>
<tr>
<td>SFNC+LF</td>
<td>Stack forcing on non-critical path and gate length biasing.</td>
</tr>
<tr>
<td>SF+LF</td>
<td>Stack forcing and gate length biasing.</td>
</tr>
<tr>
<td>VOpt+LF</td>
<td>$V_{dd}/V_{th}$ optimization and gate length biasing.</td>
</tr>
<tr>
<td>SFNC+SFNC+LF+VOpt</td>
<td>Stack forcing on non-critical path + gate length biasing + $V_{dd}/V_{th}$ optimization</td>
</tr>
</tbody>
</table>

Table 5.3. Simulation results of 65nm and 45nm technologies.

<table>
<thead>
<tr>
<th>Technology</th>
<th>65nm</th>
<th>45nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leakage Power</td>
<td>3.43e-5</td>
<td>2.53e-4</td>
</tr>
<tr>
<td>Leakage Uncertainty</td>
<td>0.38</td>
<td>0.46</td>
</tr>
<tr>
<td>Dynamic Power</td>
<td>2.74e-4</td>
<td>3.07E-04</td>
</tr>
<tr>
<td>Delay</td>
<td>2.15E-10</td>
<td>1.5E-10</td>
</tr>
<tr>
<td>Delay Uncertainty</td>
<td>0.12</td>
<td>0.23</td>
</tr>
</tbody>
</table>

5.5 Effect of Technology Scaling on Uncertainty

The power and delay simulation results of the test circuit are shown in Table 5.3 and can be used for evaluating the impacts of active leakage reduction techniques. Be reminded that throughout this chapter, the uncertainty is defined as the standard deviation divided by the mean value (S.D./mean). By comparing the data of 65nm and 45nm, we can see that both the delay and leakage power uncertainties increase with technology scaling. Note that the leakage uncertainty is larger than delay uncertainty across technologies. This is due to the stronger dependence of leakage (exponential) on threshold voltage than the dependence of delay (logarithmic) on threshold voltage.

5.6 Effect of Leakage Reducing Scheme on Uncertainty

The leakage distribution of all the evaluated techniques are plotted in Figure 5.9. The data shown is for 65nm technology. It can be seen that, for all the evaluated techniques, not only the worst case leakage is reduced but the distribution is sharper, which means better yield control. Another observation is that SF achieves most leakage savings and results in least leakage uncertainty. However,
SF is only recommended in non-critical paths due to its large delay penalty. From Figure 5.9, we can see that combining LB and SF in non-critical path (SFNC+LB) achieves additional leakage savings and provides further uncertainty reduction. SFNC+LB achieves similar leakage savings and leakage uncertainty as SLV, which includes tuning $V_{dd}/V_{th}$. However, SLV offers better delay uncertainty and lower dynamic power consumption. For designs that delay is not the primary constraint, it is recommended to use $SF+LB$ hybrid technique to control the yield while achieving minimum leakage power.

![Figure 5.9. Leakage distribution of evaluated techniques.](image)

The results of 65nm and 45nm technologies are summarized in Table 5.4 and Table 5.5, respectively. The numbers are normalized with respect to the corresponding values for the base case where no leakage reduction technique was used (Orig). All the evaluated techniques reduce dynamic/leakage power at the cost of delay penalty. Due to different optimization points for reducing leakage power and
delay/leakage uncertainty through $V_{dd}/V_{th}$ tuning, when it is tuned to minimize leakage power, the leakage and delay uncertainties could deteriorate as can be seen from Table 5.4 and Table 5.5. The lower supply voltage incurs larger leakage uncertainty while lower $V_{dd}/V_{th}$ ratio helps to reduce the leakage uncertainty. On the other hand, both conditions result in larger delay uncertainty.

<table>
<thead>
<tr>
<th>Orig</th>
<th>SF</th>
<th>LB</th>
<th>VFopt</th>
<th>SNC+LB</th>
<th>SF+LB</th>
<th>VOpt+LB</th>
<th>SLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leakage Power</td>
<td>1.00</td>
<td>0.08</td>
<td>0.14</td>
<td>0.45</td>
<td>0.14</td>
<td>0.09</td>
<td>0.04</td>
</tr>
<tr>
<td>Leakage Uncertainty</td>
<td>1.00</td>
<td>0.45</td>
<td>0.45</td>
<td>0.91</td>
<td>0.52</td>
<td>0.25</td>
<td>0.42</td>
</tr>
<tr>
<td>Dynamic Power</td>
<td>1.00</td>
<td>0.84</td>
<td>0.89</td>
<td>0.44</td>
<td>0.64</td>
<td>0.89</td>
<td>0.41</td>
</tr>
<tr>
<td>Yield (%</td>
<td>100</td>
<td>0.4</td>
<td>0.86</td>
<td>0.16</td>
<td>0.11</td>
<td>0.16</td>
<td>0.14</td>
</tr>
<tr>
<td>Delay</td>
<td>1.00</td>
<td>1.90</td>
<td>1.48</td>
<td>1.29</td>
<td>1.88</td>
<td>4.40</td>
<td>1.99</td>
</tr>
<tr>
<td>Delay Uncertainty</td>
<td>1.00</td>
<td>0.71</td>
<td>0.64</td>
<td>1.27</td>
<td>0.62</td>
<td>0.55</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 5.4. Results in 65nm technology.

<table>
<thead>
<tr>
<th>Orig</th>
<th>SF</th>
<th>LB</th>
<th>VFopt</th>
<th>SNC+LB</th>
<th>SF+LB</th>
<th>VOpt+LB</th>
<th>SLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leakage Power</td>
<td>1.00</td>
<td>0.33</td>
<td>0.08</td>
<td>0.46</td>
<td>0.03</td>
<td>0.07</td>
<td>0.02</td>
</tr>
<tr>
<td>Leakage Uncertainty</td>
<td>1.00</td>
<td>0.33</td>
<td>0.33</td>
<td>1.04</td>
<td>0.98</td>
<td>0.37</td>
<td>0.39</td>
</tr>
<tr>
<td>Dynamic Power</td>
<td>1.00</td>
<td>0.63</td>
<td>0.62</td>
<td>0.44</td>
<td>0.46</td>
<td>0.50</td>
<td>0.70</td>
</tr>
<tr>
<td>Yield (%</td>
<td>100</td>
<td>0.9</td>
<td>99.4</td>
<td>100</td>
<td>96.4</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>Delay</td>
<td>1.00</td>
<td>3.18</td>
<td>1.05</td>
<td>1.12</td>
<td>2.10</td>
<td>5.89</td>
<td>2.20</td>
</tr>
<tr>
<td>Delay Uncertainty</td>
<td>1.00</td>
<td>0.44</td>
<td>0.38</td>
<td>1.16</td>
<td>0.44</td>
<td>0.46</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Table 5.5. Results in 45nm technology.

The fifth rows of Table 5.4 and Table 5.5 show the yield achieved for each technique and can be used for evaluating the impact on the yield given the increase in delay and delay uncertainty in some of the techniques. Note that the yield presented is defined as the percentage of the cases, out of the 1000 simulation runs, in which the resulting delay is within the worst-case corner delay of Orig (shown as the point W in Figure 5.10) where no optimization technique is used. The worst-case corner delay time of Orig is the delay time assuming all the worst case conditions happened at the same time (both a 10% increase in threshold voltage and gate length). While this worst-case corner rarely happened, this is what the corner case simulation assumed in the conventional static timing analysis. The results show that increasing the gate length by 10% achieves 86% savings in leakage power and 86.6% yield. In comparison, optimizing $V_{dd}$ and threshold voltage ratio achieves 55% leakage savings and 88.5% yield. As a result, even though the average delay time increases when using leakage reduction techniques, based on corner simulation results, moderate yield can still be achieved while leakage power is reduced. We would like to point out that only the transistor delay is evaluated.

\footnote{For $VOpt$ in 65nm Technology, $V_{dd}/V_{th}$ is set to be 0.7V/0.2V.}

\footnote{For $VOpt$ in 45nm Technology, $V_{dd}/V_{th}$ is set to be 0.5V/0.16V.}
in this experiment.

To study the influence of the leakage reduction techniques on delay uncertainty, the delay distribution of each technique evaluated is plotted in Figure 5.10. Vopt technique increases the delay uncertainty while SF and LB have narrow spread of delay distribution. In Vopt, since the $V_{dd}/V_{th}$ used for the non-critical path has a higher delay uncertainty than that of the critical path in the presence of process variation, the non-critical path can actually turn out to be the delay bottleneck. Consequently, the scope for the leakage reduction using $V_{dd}/V_{th}$ optimization for non-critical paths reduces in the presence of process variation. Given the dependence between the Vopt technique and delay uncertainty, and as statistic timing analysis becomes standard to contend with the within-die variations, power optimization through $V_{dd}/V_{th}$ tuning needs to be done statistically to achieve maximum power savings while preventing the build-up of critical paths.

![Figure 5.10](image)

**Figure 5.10.** Delay time of test circuit for each technique. W point is the worst-case corner delay of Orig.

Both SF and BL reduce leakage and delay uncertainties efficiently and their effectiveness increases for smaller technologies where SCE is more pronounced. This is due to their ability to reduce the second order effect of process variations discussed in Section 5.2. According to the results, each technique is recommended for
different design considerations. Table 5.6 shows the recommended target designs.

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Target designs</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOpt</td>
<td>High performance systems where delay is constrained. Yield is traded for leakage power savings.</td>
</tr>
<tr>
<td>SFNC+LB, LB</td>
<td>Low power systems where delay is not so critical. Circuits in non-critical paths in a design.</td>
</tr>
<tr>
<td>SF, VOpt+LB, SF+LB, SIA</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6. Recommended target designs for each technique.

5.7 Temperature Variation Sensitivity

Due to the exponential dependence of threshold voltage on temperature, we also evaluate the temperature sensitivity of delay and leakage uncertainties. Monte Carlo analysis is done at temperature from 25°C to 125°C. The uncertainty shown in Figure 5.11 is the deviation in the delay/leakage divided to the mean value of leakage/delay at 25°C. The uncertainty is then normalized to the numbers of Orig for comparison. From the data shown in Figure 5.11, we can see that both delay and leakage uncertainties increase with temperature. This can be due to the increasing leakage and larger delay at high temperature. However, the behaviors of leakage uncertainty and delay uncertainty across temperature variations tend to be different across techniques. Employing LB and SF, the variation in leakage uncertainty with temperature is larger as compared to Orig and Vopt. This is because the mean leakage is lower in the cases of LB and SF as compared to Orig. However, we also see that the uncertainty of LB is larger than that of SF, whose mean leakage is smaller. This is due to the higher equivalent threshold voltage when employing SF. When applying LB, gate length is increased and thus SCE is reduced, which in turn raises the equivalent threshold voltage. For SF, the equivalent threshold voltage is further increased due to the reduced DIBL. In the case of Vopt, since the optimized condition tunes the V_{dd} only, it is similar to Orig where is no leakage reduction technique is applied. For delay uncertainty, Vopt exhibits similar behavior as Orig while LB and SF exhibit larger delay uncertainty. For SF, the delay uncertainty doubles from room temperature to 125°C. We explain the different behavior for delay uncertainty as follows. Figure 5.12 illustrates the reverse logarithmic relation between delay time and gate driving voltage. The operating range of the gate driving voltage for each technique is also shown. Vopt operates in the same range as Orig due to unchanged threshold voltage when it
applies. As a result of the above mentioned raised equivalent threshold voltage, designs with either $SF$ or $LB$ applied operate at lower gate driving voltage. As illustrated in Figure 5.12, with the same amount of variation in threshold voltage, the change in delay time of $SF$ is larger than the other techniques and this explains for its larger delay uncertainty than that of other techniques.

![Figure 5.11. Temperature sensitivity of uncertainties of the evaluated techniques.](image)

### 5.8 Conclusions

The influence of leakage reduction techniques on delay and leakage uncertainties is evaluated. It is found that the technique using lower power supply level or high $V_{dd}/V_{th}$ increases both delay and leakage power uncertainties. Special care is needed to control the yield when optimizing delay and power trade-off through tuning $V_{dd}/V_{th}$. Stack forcing and increasing gate length reduce the delay and leakage uncertainties at the cost of delay and area penalties. It is suggested that $V_{dd}/V_{th}$ tuning should be done in a statistic fashion while gate length tuning and stack forcing can be assigned with static timing analysis. To further exploit the trade-off space, we suggest some hybrid techniques. By increasing gate length of
Figure 5.12. Illustration of delay time vs gate driving voltage.

every transistor and forcing stack of transistors in non-critical paths, the delay and leakage power uncertainties can both be controlled while achieving noticeable leakage savings. Both delay and leakage uncertainties are expected to increase with technology scaling. Fortunately, the effectiveness of controlling the uncertainty through stack forcing and increasing gate length also increases. Another observation is that increasing the gate length or forcing stacked transistors, the temperature sensitivity of both delay and leakage uncertainties increase. The results presented in this chapter point out the importance of considering the delay and leakage uncertainties when applying leakage reduction techniques and quantify the uncertainties caused by process variations.
Conclusions and Future Work

6.1 Summary

Increasing leakage poses a major obstacle to continued scaling of transistors and thus draws great attention for research on leakage control, leakage avoidance, and leakage tolerance. This thesis provides a comprehensive study of leakage reduction techniques which can be used as guidelines for designers and provides a full-chip leakage simulator a framework for evaluating leakage reduction techniques for different applications. While abundant leakage reduction techniques have been proposed in literature and implemented in industry, the comparisons of these techniques and tools to estimate leakage power are valuable. The summary of the major results of this thesis is given below.

Chapter 2 provides a comprehensive evaluation of run-time leakage reduction techniques. Most run-time leakage reduction techniques are extended from three basic categories which include Input Vector Control (IVC), Body Bias Control (BBC), and Power Supply Gating (PSG). The results predict that the efficiency of BBC will reduce as technology scales while that of others increase. However, even though the effectiveness of BBC decreases, the leakage reduction will be significant (50% in average for technology in which subthreshold leakage dominates and 15% in average for technology in which gate leakage is comparable to subthreshold leakage) and the minimum idle time can be tuned to a desirable value by upsizing the drivers down to 70nm technology. Due to the increasing percentage of the leakage power, minimum idle time for overall energy to be saved for
all the techniques evaluated decreases regardless the trends of effectiveness. This promises the feasibility of these techniques even when there are less slacks of idleness resulted by the increasing operating speed. Moreover, process variations are crucial to leakage power in advanced technologies and the efficiency of BBC and PSG are more consistent than that of IVC when considering process variations.

Chapter 3 proposes leakage-aware crossbar designs for on-chip networks. Staggered buffers which assigns high-\(V_{th}\) transistors alternatively in the buffers along interconnects and pre-charge technique are used to control the leakage power in crossbar designs taking advantages of the properties of crossbar in on-chip networks. The proposed schemes saves 10.13% - 43.7% and 12.36% - 93.68% of active leakage and standby leakage, respectively, at no delay penalty. Optimizing the interconnect structure by segmenting the interconnect and properly assigning the high-\(V_{th}\) transistors reduce the leakage further by 30% more in average. The proposed schemes are also shown to be promising for a large range of flit sizes.

Chapter 4 presents an architectural level leakage power simulator to enable the exploration of various optimizations. The presented simulator considers the dynamic temperature feedback and process variations which are the parameters among that have strong impacts on leakage power. The hierarchical leakage power modeling in transistor, circuit, and architecture levels are discussed in detail in the hope of providing a framework to custom the simulator to other types of processors. The results of the simulator presented in Chapter 4 shows the importance of including temperature feedback and process variations when estimating full-chip leakage power. The experimental results show that the leakage power will exceed 50% of the chip power budget for most of the benchmarks at 65nm technology. Without using any leakage control techniques, the majority of leakage power comes from the cache hierarchies. However, the use of high threshold transistors for cache cells and applying sleep mode control while taking the temperature profiling into account, make the other processor units with higher activity no less important from a leakage power perspective. The process variation sensitivity of the leakage power has been stressed recently. Our simulation result shows that the leakage power increases by 20%-30% at architecture level when accounting for nominal process variations in gate length, oxide thickness, and channel doping concentration.

Chapter 5 examines the influence of circuit level leakage reduction techniques
on delay/leakage uncertainties. The techniques studied include $V_{dd}/V_{th}$ tuning, gate length biasing, stacking forcing and hybrid techniques using the above mentioned techniques. It is found that the technique using lower power supply level or high $V_{dd}/V_{th}$ increases both delay and leakage power uncertainties. Stack forcing and increasing gate length reduce the delay and leakage uncertainties at the cost of delay and area penalties. By increasing gate length of every transistor and forcing stack of transistors in non-critical paths, the delay and leakage power uncertainties can both be controlled while achieving noticeable leakage savings. The impacts of technology scaling are also studied. Both delay and leakage uncertainties are expected to increase with technology scaling. Fortunately, the effectiveness of controlling the uncertainty through stack forcing and increasing gate length also increases. Another observation is that increasing the gate length or forcing stacked transistors, the temperature sensitivity of both delay and leakage uncertainties increase. The results presented in this chapter point out the importance of considering the delay and leakage uncertainties when applying leakage reduction techniques and quantify the uncertainties caused by process variations and thus are important for the leakage and uncertainty co-optimizations.

6.2 Future Work

Development of technology improvement for mitigating leakage power is thriving. Nevertheless, to maintain the power budget while achieving high performance, solutions that combine optimizations at the architecture level and circuit design level are necessary. While this thesis provides in detail the discussion of circuit level leakage reduction techniques and proposes an architecture level full-chip leakage simulator, it would be enlightening to integrate the circuit level techniques into the architecture level simulator so that the overall power profiling can be generated. This overall power profiling is essential for evaluating optimizations at every level such as circuit, compiler, and architecture levels.

This thesis proposes leakage-aware crossbar schemes that achieve significant standby leakage savings. However, to mitigate the switching power penalty due to pre-charge, coding schemes that provide the least switching activity while achieving biased state probability can further enhance the total power savings.
Another technology scaling obstacle coming with increasing leakage power is the thermal issue. Thus, thermal issues for future designs are also of interests. Interconnects dominate the performance and power behavior of deep submicron designs. Consequently, interconnect centric design methods and technology improvements are critical to the chip industry. While there have been significant interconnect technology improvements over the last few years such as the use of copper and low-K dielectric, the industry is striving for additional improvements. The various technologies being actively explored to address the interconnect problem include the use of packet-based on chip communication networks [76], use of angular wires instead of Manhattan routing [77], on-chip optical interconnect[79], and the use of three-dimensional chips [78]. A three dimensional (3D) chip is a stack of multiple device layers with direct vertical interconnects tunneling through them. A key benefit of this approach over a traditional two dimensional chip is the ability to reduce the length of long interconnects. A tool estimating performance and power of 3D cache is proposed in [80] and the initial results show significant performance improvement can be achieved by 3D structures. However, the elevated temperature/thermal caused by stacking of chips worsens the leakage power the offsets the savings in dynamic power. Thus techniques for mitigating thermal and leakage problems of 3D ICs are considered exciting research topics.
Abbreviations

For sake of convenience, the abbreviations used in this thesis is listed in the next page.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSFET</td>
<td>Metal Oxide Semiconductor Field Emission transistor</td>
</tr>
<tr>
<td>NMOS</td>
<td>N-type Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>PMOS</td>
<td>P-type Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>$V_{th}$</td>
<td>Threshold Voltage</td>
</tr>
<tr>
<td>SCE</td>
<td>Short Channel Effect</td>
</tr>
<tr>
<td>DIBL</td>
<td>Drain-Induced Barrier Lowering</td>
</tr>
<tr>
<td>FN</td>
<td>Fowler-Nordheim</td>
</tr>
<tr>
<td>BPTM</td>
<td>Berkeley Predictive Technology Model</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>6T cell</td>
<td>six-transistor cell</td>
</tr>
<tr>
<td>MTCMOS</td>
<td>Multi-Threshold Complementary Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>DTMOS</td>
<td>Dynamic Threshold Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>VTCMOS</td>
<td>Variable Threshold Complementary Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>SOI</td>
<td>Silicon on Insulator</td>
</tr>
<tr>
<td>IVC</td>
<td>Input vector Control</td>
</tr>
<tr>
<td>BBC</td>
<td>Body Bias Control</td>
</tr>
<tr>
<td>PSG</td>
<td>Power Supply Gating</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase-Locked Loop</td>
</tr>
<tr>
<td>VLIW</td>
<td>Very Long Instruction Word</td>
</tr>
<tr>
<td>TLB</td>
<td>Table Look-aside Buffer</td>
</tr>
<tr>
<td>FUB</td>
<td>Functional Unit Block</td>
</tr>
<tr>
<td>CAM</td>
<td>Content Addressable Memory</td>
</tr>
<tr>
<td>NoC</td>
<td>Networks-on-Chip</td>
</tr>
<tr>
<td>DFC</td>
<td>Dual-Threshold Feedback Crossbar</td>
</tr>
<tr>
<td>DPC</td>
<td>Dual-Threshold Pre-charge Crossbar</td>
</tr>
<tr>
<td>SDFC</td>
<td>Segmented Dual-Threshold Feedback Crossbar</td>
</tr>
<tr>
<td>SDPC</td>
<td>Segmented Dual-Threshold Pre-charge Crossbar</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
</tbody>
</table>
Bibliography


[56] Liao, W.; Fei Li; and Lei He; "Microarchitecture Level Power and Thermal Simulation Considering Temperature Dependent Leakage Model", International Symposium on Low Power Electronics and Design, Aug 2003, pp. 211-216


[63] BSIM4.2.1 MOSFET Model. http://www-device.eecs.berkeley.edu/bsim3


Vita

Yuh-Fang Tsai

Yuh-Fang Tsai received the B.S. degree in Electronics Engineering from Chun-Yuan Christine University, Chun-Li, Taiwan in 1996. In 2002, she received the M.S. degree in Computer Science and Engineering from the Pennsylvania State University, University Park, PA. Her research interests include leakage power management and power aware VLSI circuit and system designs.