MicroFix: Using Timing Interpolation and Delay Sensors for Power Reduction

GUIHAI YAN, YINHE HAN, and HUI LIU, Chinese Academy of Sciences
XIAOYAO LIANG, NVIDIA Corporation
XIAOWEI LI, Chinese Academy of Sciences

Traditional DVFS schemes are oblivious to fine-grained adaptability resulting from path-grained timing imbalance. With the awareness of such fine-grained adaptability, better power-performance efficiency can be obtained. We propose a new scheme, MicroFix, to exploit such fine-grained adaptability. We first show the potential resulted from the path-grained timing imbalance and then present a new technique, Timing Interpolation, to reap the fine-grained adaptability for power reduction. Moreover, to eliminate the conservative margins of traditional DVFS, unlike the previous approaches such as Razor that reactively handle the delay errors (induced by aggressively scaled voltage/frequcncy) by enabling error detection and recovery, we propose a proactive approach by error prediction, thereby obviate the high-cost recovery routines. MicroFix was evaluated based on ISCAS89 benchmarks and the floating-point unit adopted by OpenSPARC T1 processor. Compared to ideal traditional DVFS schemes, the experimental results show that for most of the evaluated circuits, MicroFix can help saving up to 20% power consumption without compromising with frequency, at the expense of less than 5% area overhead. Compared to nonideal DVFS schemes (with 10% voltage margin), the power reduction can even reach up to 38% on average.

Categories and Subject Descriptors: B.7 [**Integrated Circuits**]: General

General Terms: Design, Performance, Reliability

Additional Key Words and Phrases: Power reduction, fine-grained adaptability, DVFS, timing interpolation, delay sensor

ACM Reference Format:
Yan, G., Han, Y., Liu, H., Liang, X., and Li, X. 2011. MicroFix: Using timing interpolation and delay sensors for power reduction. ACM Trans. Des. Autom. Electron. Syst. 16, 2, Article 16 (March 2011), 21 pages.
DOI = 10.1145/1929943.1929948 http://doi.acm.org/10.1145/1929943.1929948

1. INTRODUCTION

The objective of dynamic voltage/frequency scaling (DVFS) [McGowen et al. 2006; Herbert and Marculescu 2007] is to achieve an optimal efficiency, that is, power-performance tradeoff, though the efficiency may imply different preference between power and performance [Brooks et al. 2000]. The low-power, portable systems

A preliminary version of the article appeared as Yan et al. [2009b]. This article extends the preliminary version by introducing a new model to guide the optimum design and more comprehensive evaluation to the proposed scheme.
The work was supported in part by National Basic Research Program of China (973) under grant No. 2011CB302503; in part by National Natural Science Foundation of China (NSFC) under grants No. 60806014, 61076037, 60906018, 60921002, 60831160526, and 60633060; and in part by Hi-Tech Research and Development Program of China (863) under grant No. 2009AA01Z126.
Authors' addresses: G. Yan, Y. Han, H. Liu, and X. Li, Key Laboratory of Computer System and Architecture, ICT, CAS, Beijing, China; email: {yan_guihai, yinhes, liuhui, lxw}@ict.ac.cn.; X. Liang, NVIDIA Corporation; email: xliang@nvidia.com.

DOI 10.1145/1929943.1929948 http://doi.acm.org/10.1145/1929943.1929948

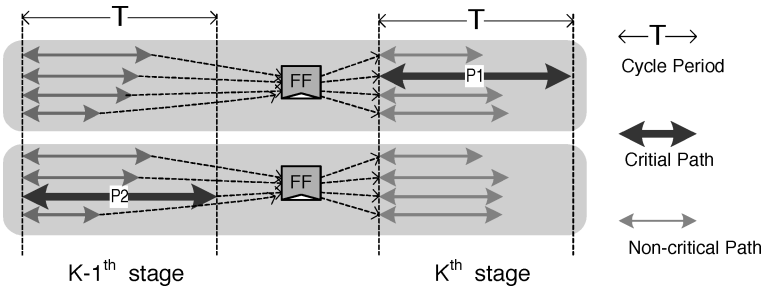


Fig. 1. Impact of DVFS to path delay.

emphasize more power than performance while the high-performance-oriented systems take the opposite approach. However, traditional DVFS schemes are bound to be far from optimum for two reasons.

First, most of the traditional DVFS schemes such as Intel's SpeedStep [Intel 2004] and AMD's PowerNow [AMD 2000] technologies are operated in chip level or core level [Isci et al. 2006; Dorsey et al. 2007]. Such coarse-grained approaches are naturally oblivious to fine-grained adaptability that can be exploited to accommodate more pertinently the potential delay errors caused by reduced voltage or increased frequency. The traditional voltage-frequency tradeoffs are usually determined by the longest paths. Figure 1, for example, shows two stages of a pipeline after voltage scaling down; in each stage there is a critical path ($P1$ in K^{th} stage and $P2$ in $K-1^{th}$ stage) that can cause delay errors if no corresponding frequency scaling involved. This fact is derived from a coarse-grained perspective. However, from a fine-grained, that is, path-grained, perspective such potential delay error can be eliminated without sacrificing the frequency as long as intentional time stealing is conducted, since for $P1$ all of the related upstream paths are timing noncritical, and for $P2$ all of the downstream paths are also timing noncritical. Such path-timing imbalance, called intrinsic imbalance, are very common in current circuits (Section 2). Furthermore, we find that given a constant frequency, voltage scaling down can exacerbate the path-grained timing imbalance. We aim to fully exploit such path-grained timing imbalance to enable power reduction but without compromising with performance, thereby achieving a better power-performance tradeoff.

Second, the traditional DVFS operations are predetermined, hence can not adapt to individual chips that may slightly differ from each other. With the soaring process variation, however, the batch of chips tagged even in the same frequency bins are spread out in a narrow frequency band. Hence, even a set of predetermined Voltage-Frequency configurations is optimum for some corner chips, the same configurations are doomed to be nonoptimum for the others. Actually, previous research has evidenced that such nonadaptive DVFS approaches always tend to be overconservative [Das et al. 2006] to ensure safe timing in the worst case, thereby wasting considerable power consumption.

We propose an adaptive path-grained approach, called MicroFix [Yan et al. 2009b], to enhance the traditional DVFS schemes by overcoming the above two limitations. Given a chip, the adaptability is realized by introducing a set of delay sensors [Yan et al. 2009a] used to indicate the "just-enough" voltage level. During the voltage scaling down operations, when one or multiple of the delay sensors tell timing violations are impending, the voltage scaling stops immediately and is followed by restoring a tight margin, thereby reaching a just-enough voltage level. The proposed MicroFix inherits all the benefits brought by Razor [Das et al. 2006], but in another way: error prediction, rather error detection and recovery.

In terms of MicroFix implementation, one essential problem is how to determine the skews of each flip-flop to maximally exploit the path-grained adaptability. We propose a new technique, Timing Interpolation, to tackle this problem. First, we set up an optimization problem to maximize the slack of timing imbalanced paths; then we group the solution of this problem by using K-means clustering procedure. Each cluster corresponds to an interpolating clock. We find that usually $1\backsim2$ extra clocks can fully exploit the adaptability.

We have evaluated MicroFix based on ISCAS89 circuit benchmarks and an industry floating point unit (FPU). The experimental results show that for most of the evaluated circuits, MicroFix can help saving up to 20% power consumption without compromising with frequency, at the expense of less than 5% area overhead. Compared against non-ideal DVFS schemes (with 10% voltage margin), the power reduction can even reach up to 38% on average.

The rest of this article is organized as follows: Section 2 investigates the related works. Section 3 shows the timing imbalance of the FPU and describes the effect of MicroFix on DVFS. Section 4 presents the MicroFix enhanced DVFS architecture. Section 5 addresses implementation details. Section 6 evaluates the effectiveness of MicroFix, followed by conclusion in Section 7.

2. RELATED WORK

The proposed MicroFix scheme in essence is timing stealing. Traditionally, the timing adaptation for a pipeline is conducted at coarse-grained, that is, stage-grained. ReCycle [Tiwari et al. 2007] is a typical example using the stage-grained time borrowing for tolerating process variation. Such stage-grained approaches, however, just can exploit limited adaptability, and are even rendered ineffective for well balanced pipelines (the pipeline stages has the equal delay). Unlike those coarse-grained approaches, MicroFix, which relies on a path-grained timing adaptation mechanism, suffers little from the balanced pipelines. This discovery offers an new opportunity for reducing the voltage level more aggressively.

Aggarwal et al. proposed an adaptive on-chip voltage regulation scheme to provide the optimum voltage level to the target circuits [Aggarwal and Carley 1999; Dragone et al. 2000]. This scheme uses the delay of critical path replica to indicate the just-enough voltage of the target circuits. However, given the increasing process variation, especially the random component, the critical path replica may not be competent to serve as the timing indicator any more. Hence, the replica indicated voltage level, though still be adaptive, may not the optimum to the host circuit.

Ernst et al. proposed the Razor [Ernst et al. 2003; Das et al. 2006] to eliminate the conservativeness of voltage margin. Unlike the prior adaptive on-chip voltage regulation scheme [Dragone et al. 2000], Razor does not rely on any circuit replica but directly handles critical paths. The sink flip-flops of critical paths are replaced with a new type of flip-flops that are equipped with delay error detection capability; the target pipeline is designed to be able to recover from delay errors. With such reliability-ensuring remedy, the conservative voltage margin can be safely minimized. Razor can bring two benefits: 1) eliminating the conservativeness of voltage and 2) adapting to different chips suffered from process variation.

Similarly to Razor, MicroFix takes the delay errors as the DVFS indicator. The difference from Razor lies in the way the delay errors are dealt with: Razor handles the delay errors in a reactive manner—detection and then recovery, while MicroFix handles delay errors in a proactive manner—prediction. The "proactive" MicroFix obviates the recovery logics, while the prediction logics (delay sensors) are as cost-efficient as detection logics of Razor. Our previous work [Yan et al. 2009a] proposed an efficient

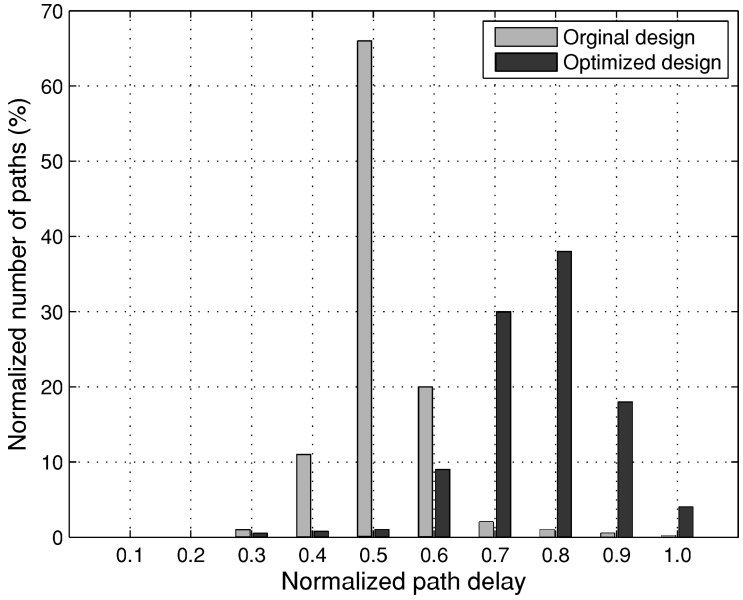


Fig. 2. Path-delay distribution in the original and optimized design (Adapted from Usami et al. [1998]).

delay sensor design which can be used in MicroFix scheme.The following gives a brief description of the sensor design.

Basically, a delay error of a signal can be reflected as unintentional signal transitions in a specified period of time during which the signal should have reached a stable state. The unintentional transitions can be modeled as stability violations [Yan et al. 2009a]. Agarwal et al. [2007] used such stability violation for predicting aging induced delay errors. The same strategy can be also used to predict the delay errors induced by voltage reduction or frequency increase. For more details about sensor circuit design, please refer to Yan et al. [2009a].

3. PATH-GRAINED TIMING IMBALANCE

First, to get a feel for the abundance of path delay imbalance, we represent the path delay distribution of a media processor chip provided by Usami et al. [1998]. Figure 2 shows that the intrinsic path delay imbalance is very common, even for the optimized design. Hence, we believe that the path-grained adaptability can provide new potential to power reduction.

We focus on the imbalance that can be exploited for improving DVFS efficiency. Such timing imbalance can be reflected by characterizing the flip-flops. A flip-flop can be categorized according to the slack values of related upstream and downstream paths. Specifically, suppose a flip-flop FF serves as the end point of m paths with slack values $e_1, e_2, \ldots, e_m$, and the start point of n paths with slack values $s_1, s_2, \ldots, s_n$. Given a threshold, TH, which distinguishes the (potential) critical paths ($slack \leq TH$) from others ($slack > TH$), the flip-flop must fall into one of the four classes.

—Generous Flip-flop (GFF). All of the $\{e_1, e_2, \ldots, e_m\}$ and $\{s_1, s_2, \ldots, s_n\} > TH$. ("Generous" with time margin.)
—Backward Adaptable Flip-flop (BAFF). At least one of the $\{e_1, e_2, \ldots, e_m\} \leq TH$, but all of the $\{s_1, s_2, \ldots, s_n\} > TH$.

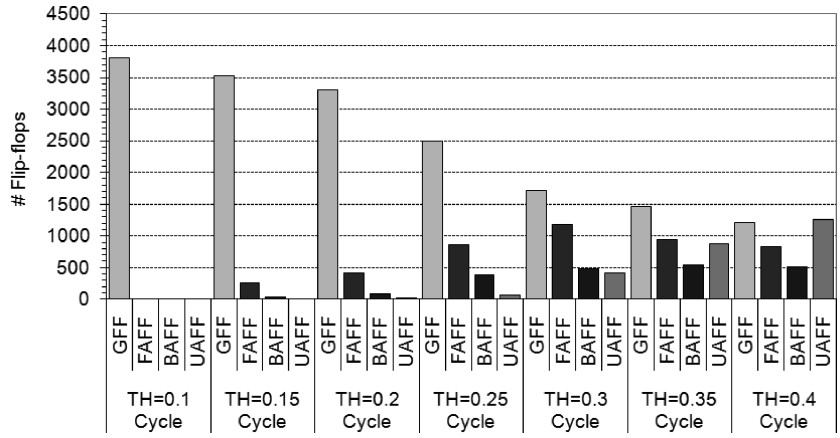


Fig. 3. Distribution of flip-flops with different *TH*.

—Forward Adaptable Flip-flop (FAFF). All of the $\{e_1, e_2, \ldots, e_m\} > TH$, but at least one of the $\{s_1, s_2, \ldots, s_n\} \leq TH$.
—Unadaptable Flip-flop (UAFF). At least one of the $\{e_1, e_2, \ldots, e_m\} \leq TH$, and at least one of the $\{s_1, s_2, \ldots, s_n\} \leq TH$.

Note that besides GFFs, FAFFs, and BAFFs can also serve as margin provider, though in a "unidirectional" manner. In addition, a path may start and end with the same flip-flop, that is, a path in a tight loop [Borch and Tune 2002], but these rules are still valid.

Threshold *TH* is a key factor affecting the design trade-offs. On one hand, the larger *TH*, the higher the percentage of UAFFs, thus the less efficiency improvement that MicroFix can provide because UAFFs and related critical path isolation, referred to "critical isolation" in the following, are handled by conservative DVFS; on the other, the larger *TH* can facilitate the implement of more aggressive DVFS with MicroFix on the noncritical portion. Section 6 will discuss the detailed implication of *TH*.

Since the efficacy of MicroFix is derived from the timing imbalance, in the following we first discuss the path-grained timing imbalance from two aspects: 1) intrinsic imbalance and 2) how does DVFS affect such imbalance, and then show how such imbalance facilitates a more efficient DVFS scheme.

3.1. Intrinsic Imbalance

As a case study, we choose a pipelined FPU adopted by Open-SPARC T1 [Sun Microsystem Inc. 2006] processor as our target pipeline which implements the SPARC V9 floating-point instructions and supports all IEEE 754 floating-point data types. This FPU is synthesized using Synopsys Design Compiler with UMC 0.18um technology. We set the performance as the synthesizing priority to smooth the distribution of path delay as much as possible. Then, we analyze the path timing with PrimeTime. The cycle period is defined as the critical path delay plus 10% time margin (in fact 10% margin is quite tight in the deep submicro technology).

Figure 3 shows the number of flip-flops with different type of adaptability under seven *TH* configurations: 0.1 (cycle), 0.15, 0.2, 0.25, 0.3, 0.35, and 0.4. All of the flip-flops are "generous" when $TH = 0.1$ because 10% margin are left. With *TH* increasing, some GFFs fall into the groups of BAFFs or FAFFs, even UAFFs. Note that the BAFFs and FAFFs, which also serve as the bottlenecks of traditional DVFS approaches, always take considerable proportions. MicroFix can relax such bottlenecks by enabling path-grained

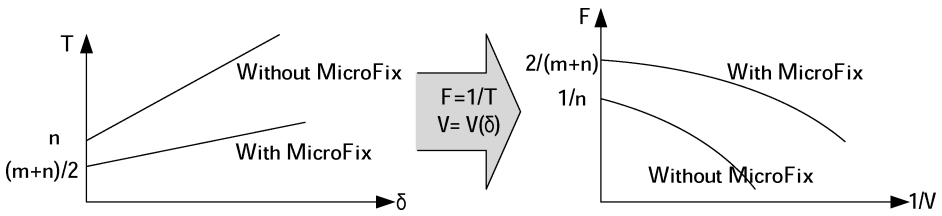


Fig. 4. Effect of MicroFix on DVFS.

time stealing; thereby achieving the objective: trading less frequency degradation with the same voltage reduction, or equivalently, obtaining more voltage reduction with the same frequency degradation.

Furthermore, reducing voltage results in different delay change for different paths. Generally, the longer paths suffer more delay increase than the shorter ones.

These observations motivate us to exploit the imbalance to improve the traditional DVFS schemes.

3.2. Exploiting Timing Imbalance

Suppose there are two paths, P_a and P_b, in two consecutive stages of a pipeline and the end point of P_a also serves as the start point of P_b. Path P_a and P_b consist of m and n gates, respectively; without loss of generality, suppose $m < n$.

Generally, when reducing power, reducing frequency is prior to voltage scaling down to avoid hazards. Consider a step of DVFS operation, the frequency has just scaled from F $(1/T)$ down to F' $(1/T')$ then followed by corresponding voltage scaling.

From the traditional DVFS perspective, the lower bound of T' should be the largest path delay, that is,

$$T_1' = n(t_G + \delta), \tag{1}$$

where δ denotes the delay increase of a gate suffers from the voltage scaling down. Note that, as a first-order model, we suppose the delay of each gate is t_G and each gate suffers equal increase in delay after the same voltage reduction. For simplicity, we omit the trivial margin term.

From MicroFix's perspective, the lower bound of T' can be reduced to

$$T_2' = \frac{m+n}{2}(t_G + \delta) \tag{2}$$

as long as the clock of the center flip-flop of the two paths are forward skewed by

$$T_2' - m(t_G + \delta) = \frac{n-m}{2}(t_G + \delta). \tag{3}$$

Equation (3) tells how to interpolate the clock to exploit the timing imbalance. Take a BAFF as an example; although the forward slack, for the worst case, is close to zero, the backward slack is at least TH. Hence, reaping the new timing margin by skewing the clock backward by $TH/2$ allows more voltage scaling down without compromising with frequency tuning.

Because $T_2' < T_1'$ (since $m < n$), less performance degradation is imposed. Based on Equations (1) and (2), we present the qualitative effect of MicroFix in Figure 4. The parameter δ nonlinearly depends on voltage V, which can be obtained from evaluation results (Section 6). In general, MicroFix can slow down the degradation of frequency in DVFS operations. The next section presents the details of MicroFix architecture.

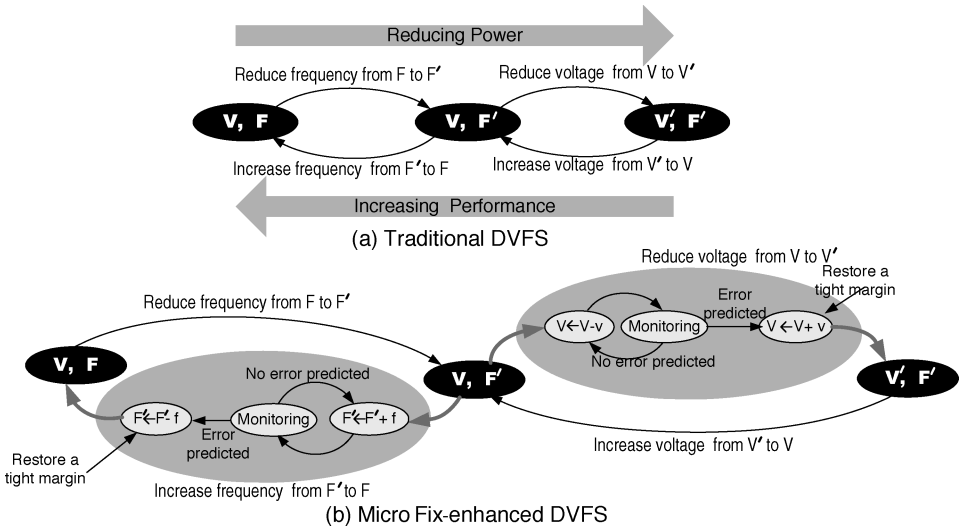


Fig. 5. MicroFix-enhanced DVFS.

4. MICROFIX-ENHANCED DVFS ARCHITECTURE

This section first presents the operational principle of MicroFix, and then describes Timing Interpolation technique employed by MicroFix. Finally, we clarify a sensor deployment policy to ensure safe tuning operations.

4.1. Operational Principle

We use delay sensors to facilitate eliminating conservative voltage/frequency margins. Previously, the sampling results of temperature sensors and power sensors have been used as the feeds of DVFS control units [McGowen et al. 2006]. Such sensors trigger the DVFS units to tune voltage-frequency to a predefined level. Although the offline determined voltage-frequency configurations are effective, they have been proved to be quite conservative [Das et al. 2006] since these configurations do not directly target delay errors which serve as a more essential factor than temperature or power to DVFS operations. Hence, MicroFix employs a group of delay sensors as the indicator to voltage reduction to eliminate the conservative voltage margin, thereby obtaining a more aggressive power reduction.

Figure 5 shows the operational state transitions. Traditionally, to reduce power, the first step is scaling down frequency from F to F', then followed by voltage reduction from V to V', as Figure 5(a) shows; to improve performance, the voltage get increased first, then followed by increase in frequency. For MicroFix-enhanced DVFS, the tuning operations get changed in two suboperations that can potentially induce delay errors: 1) reducing voltage and 2) increasing frequency. The modified suboperations, as shown in Figure 5(b), can be explained with the following steps. First, the delay sensors are turned on before conducting the "risky" suboperations; second, the voltage/frequency control unit progressively reduces the voltage (or increases frequency) under the monitoring of delay sensors till any delay error is predicted; third, return the voltage (or frequency) to a safe level by restoring a tight margin v (or f); finally, turn the delay sensors off. The tuning step v (or f) depends on T_{GB}, the width of guard band [Agarwal et al. 2007] of delay sensors.

The following presents several equations to describe some parameters of MicroFix. Suppose the resolution of voltage scaling is R_V and frequency resolution is R_F. Then

$$\delta_V = f(V, R_V) \text{ and } \delta_F = g(F, R_F), \tag{4}$$

where δ_V and δ_F denotes the delay change caused by one step of (atomic) voltage and frequency scaling, respectively. Since $T = 1/F$, we have $\delta_F = 1/F - 1/(F + R_F)$. Usually, because $R_V \ll V$ and $R_F \ll F$, we have

$$\delta_F \approx \left(\frac{1}{F}\right)^2 \times R_F. \tag{5}$$

The circuit-specific relation between voltage and delay, described as $D = f_{V \to D}(V)$, can be obtained by regression analysis of Hspice simulation results. A alpha-power law model was also proposed to model this relation [Sakurai and Newton 1990]. Generally, $D \propto (V - V_{th})^{-\gamma}$, where V_{th} is the threshold voltage and γ is a constant between 1 and 2. Then, assume $D + \delta_V \propto (V - R_V - V_{th})^{-\gamma}$, we have

$$\delta_V \propto \frac{\gamma}{(V - V_{th})^{\gamma+1}} \times R_V. \tag{6}$$

As design constrains, to avoid the delay change induced by one step voltage (or frequency) tuning running out of the prediction capability, R_V, R_F, and T_{GB} should be calibrated to meet the following relation:

$$\delta_V < T_{GB} \text{ and } \delta_F < T_{GB}. \tag{7}$$

Usually, the tight margin v and f also meet $v \ll V'$ and $f \ll F$. Hence, the restored margin v and f should satisfy

$$f(V', v) > T_{GB}, \text{ and } g(F, f) > T_{GB} \tag{8}$$

to safely accommodate the predicted delay errors.

Equation (7) and Equation (8) can be easily satisfied. For example, for a 15∼20 Fo4 pipeline fabricated using 45nm technology, 10mV of R_V, which can actually be hundreds of microvolt magnitude [McGowen et al. 2006], results in one-step delay change (δ_V) about 20ps. R_F is only required to be 10-MHz to achieve 10ps delay change (δ_F) for a 1-GHz pipeline. While MicroFix adopting the T_{GB} setting of 5% of clock period, which results in 50ps guard band (T_{GB}), is very practical [Agarwal et al. 2007].

4.2. Timing Interpolation: A Technique to Exploit Path-Grained Adaptability

In Section 3, we have qualitatively revealed the observation of timing imbalance. In this section, we propose a new technique, called Timing Interpolation (TI), to reap the fine-grained adaptability. TI is a presilicon technique for reallocating slack between timing-imbalanced paths. The timing-imbalance, if properly exploited, can be used to facilitate more aggressive voltage reduction (or frequency increase). The basic idea of TI is driving the flip-flops whose clocks need to be skewed by the same degree with the same clock, referred to interpolation clock. Hence, the primary goal is to maximize the timing margins with respect to power consumption. This problem can be modeled as follows:

4.2.1. Modeling the Problem. Suppose a circuit consists of n flip-flops and m timing paths, denoted by $F = \{f_1, f_2, \ldots, f_n\}$ and $P = \{p_1, p_2, \ldots, p_m\}$ with slack values $S = \{s_1, s_2, \ldots, s_m\}$. For simplicity, suppose there are no multi-cycle paths in the circuit. For flip-flop f_i with u upstream flip-flops $U = \{f_{i_{up,1}}, f_{i_{up,2}}, \ldots, f_{i_{up,u}}\}$ and v downstream flip-flops $V = \{f_{i_{dn,1}}, f_{i_{dn,2}}, \ldots, f_{i_{dn,v}}\}$. The longest timing paths associated with f_i and each upstream and downstream flip-flop are denoted by $P_{i,up} = \{p_{i_{up,1}}, p_{i_{up,2}}, \ldots, p_{i_{up,u}}\}$

and $P_{i,dn} = \{p_{i_{dn,1}}, p_{i_{dn,2}}, \ldots, p_{i_{dn,v}}\}$, respectively. Clearly, we have $U \subset F$ and $V \subset F$, $P_{i,up} \subset P$ and $P_{i,dn} \subset P$.

Furthermore, suppose that timing interpolation is conducted on the ith flip-flop, and the skew is denoted by δ_i. Specifically,

—if $\delta_i = 0$, no clock manipulation;
—if $\delta_i > 0$, forward skew the clock by δ_i; in contrast
—if $\delta_i < 0$, backward skew by δ_i.

After timing interpolation, the upstream slack values of f_i are

$$Slack_{i,up} = \{s_{i_{up,1}} + \delta_{i_{up,1}}, s_{i_{up,2}} + \delta_{i_{up,2}}, \ldots, s_{i_{up,u}} + \delta_{i_{up,u}}\} - \delta_i, \tag{9}$$

and the downstream slack values are

$$Slack_{i,dn} = \{s_{i_{dn,1}} - \delta_{i_{dn,1}}, s_{i_{dn,2}} - \delta_{i_{dn,2}}, \ldots, s_{i_{dn,v}} - \delta_{i_{dn,v}}\} + \delta_i. \tag{10}$$

Intuitively, given a specified slack threshold TH, the key problem is to search for an interpolation configuration $\Delta = \{\delta_1, \delta_2, \ldots, \delta_n\}$ so that as many as possible flip-flops can be eligible to be GFFs, that is

$$\begin{aligned} &\min\{Slack_{i,up}\} > TH \text{ and} \\ &\min\{Slack_{i,dn}\} > TH, \text{ for } i = 1, 2, \ldots, n. \end{aligned} \tag{11}$$

Those flip-flops that can not meet the above requirement under this Δ, including the associate paths whose slack is less than TH, have to be precluded from the application scope of MicroFix. These portions, called Critical Isolation, should be dealt with a conservative voltage level (we will discuss more in Section 5).

Unfortunately, the above problem is not a standard optimization problem and solving it is not trivial because the procedure of seeking an optimum Δ inevitably involves solving a large-scale SAT-like problem: Step 1: Set a tentative initial solution Δ_{ini}; Step 2: Verify the satisfiability of Equation (11); Step 3: Update the solution with some sophisticated heuristics in the expectation of reaching a better results. What is worse, Δ is not binary but arbitrary, which significantly extends the feasible solution space and thereby slims down the chance of reaching an even close-to-optimum solution.

We tackle this problem from another perspective. Since TI aims to maximize the number of flip-flops with a specified timing margin, we first strive to maximize the harmonic average slack of the paths, and then preclude the Critical Isolation according to TH. Therefore, the key problem is transformed to

$$\text{maximize } Slack_{avg} = \frac{1}{\sum_{i=1}^{m} \frac{1}{s_i'}}, \ s_i' > 0, \tag{12}$$

where s_i' is the slack value of the ith timing path that starts from f_j and ends with f_k. Hence

$$s_i' = s_i + \delta_j - \delta_k. \tag{13}$$

The objective $Slack_{avg}$ clearly is a function of $\Delta = \{\delta_1, \delta_2, \ldots, \delta_n\}$, $|\delta_i| < T$, $i = 1, 2, \ldots, n$.

The above problem is a standard optimization problem and can be readily solved with some mathematical tools such as MATLAB optimization toolbox. The mathematical details are omitted in this article.

4.2.2. Clustering the Solution. The given solution Δ, though can maximize the margins in the sense of harmonic mean, may be unattainable because, in such an "ideal" case, each basic block may need an individual clock that is required to be delicately skewed

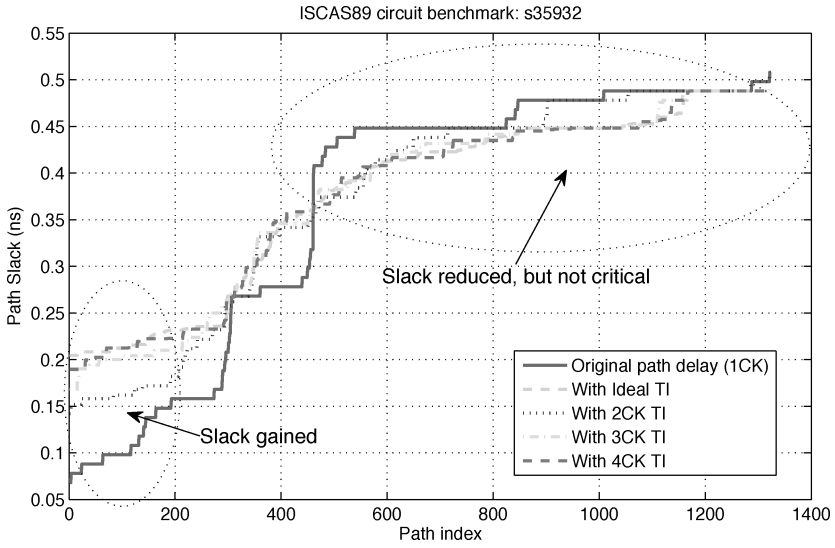


Fig. 6. The impact of TI on distribution of path delay.

from the others. Given a limited number of interpolation clocks, we have to cluster these skew values around several "representative" values that yield close-to-optimum harmonic mean of slack. This optimization can be modeled with the well-known *K-Means Clustering* problem. The number of clusters "K" is actually equal to the number of available interpolation clocks.

Intuitively, if an approximate solution δ' is "closer" to δ, the achieved slacks generally should be also closer-to-ideal. The property of "close" can be characterized by using euclidian distance. Suppose there are n elements in Δ and k interpolation clocks available, then the optimization problem can be defined as follows:

Given n skew values $\Delta = \{\delta_1, \delta_2, \ldots, \delta_n\}$, we aim to partition the n values into k sets $(k < n)$ $S = \{s_1, s_2, \ldots, s_k\}$ so as to minimize the within-cluster sum of distance; that is,

$$\text{minimize} \sum_{j=1}^{k} \sum_{\delta_i \in s_j} d(\delta_i, \mu_j), \tag{14}$$

where μ_j is the arithmetic mean of s_j.

The K-Means Clustering problem is NP-hard. Many heuristic algorithms have been proposed to effectively solve this problem. We use an algorithm in Elkan [2003].

4.2.3. Identifying Critical Isolation. TI can redistribute the path delay of the target circuit, but can not assure every path gains enough slack. After TI, the paths that fail to gain enough slack, that is, slack less than TH, and associated sink flip-flops have to be identified as critical portions, referred to Critical Isolation.

4.2.4. Example of the Impact of TI. As an example, Figure 6 shows the resultant path delay distribution for a ISCAS89 circuit benchmark s35932 under different TI configuration, where we configured TI with 2, 3, 4, and arbitrary large number of interpolating clocks, denoted by 2CK, 3CK, 4CK, and Ideal, respectively. The insights conveyed by this results can be discussed from two aspects.

First, TI can help some paths that are initially critical gain extra slack, as the left oval indicates. With 3CK TI configuration, for example, about 30% top critical timing paths can gain up to 0.2ns slack—40% of critical path delay.

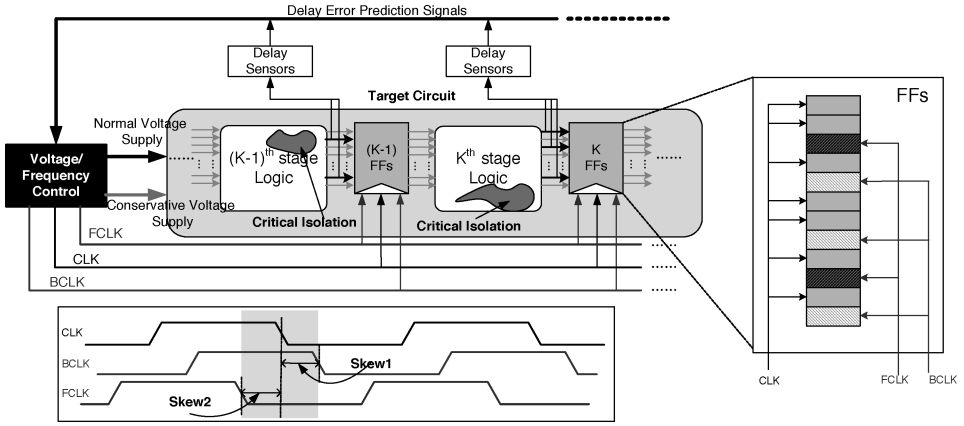


Fig. 7. Exemplify the implementation of MicroFix scheme with three interpolation clocks.

Second, the more clocks adopted, the more slack obtained; however, the marginal slack improvement declines with the increase in the number of clocks. As Figure 6 indicates, from 2CK to 3CK, a significant slack improvement on average can be achieved, but from 3CK to 4CK, even 4CK to Ideal, the slack improvement is much less prominent than that from 2CK to 3CK. Therefore, three clocks are highly recommended (which only needs two extra clocks).

In addition, we would like to clarify that some critical paths have no chance to gain extra slack even with ideal TI, though this unfavorable case seems not prominent for s35932. These paths hence have to be identified as Critical Isolation and governed with a conservative voltage level during DVFS operations.

4.3. Deployment of Delay Sensors for Tuning Safety

The safety during the two "risk" operations is guaranteed by a set of properly deployed sensors. After timing interpolation, only a subset of flip-flops need to be monitored with delay sensors. Specifically, those timing paths whose slack meets

$$TH < slack < TH + T_{GB} \tag{15}$$

will bear the brunt of the delay errors during DVFS operations. T_{GB} is determined by one-step voltage (or frequency) tuning as Equation (7) indicates.

In contrast, the other portions either reside in Critical Isolation ($slack \leq TH$) or hold relatively large margins ($slack \geq TH + T_{GB}$), and therefore are free from delay errors. Hence, after timing interpolation, those flip-flops whose data input pins are associated with at least one timing path whose slack meets Equation (15) need to be monitored with delay sensors.

5. IMPLEMENTATION OF MICROFIX

This section first presents the top view of MicroFix implementation, and then addresses three implementation issues.

Figure 7 exemplifies the top view of MicroFix scheme with three interpolation clocks. The main differences from the traditional DVFS scheme come from three aspects: 1) two extra clocks are used to reap the path-grained adaptability; 2) delay sensors are employed to enable a self-tuning DVFS; 3) two voltage supplies are deployed: the conservative one, which provide the voltage supply for the Critical Isolation illustrated as the nonregular dark shadows, is controlled by traditional DVFS, and the normal one, which is for the rest outside of the Critical Isolation, is controlled by MicroFix.

5.1. Decoupling the Critical Isolation

MicroFix scheme can not benefit from Critical Isolation. Fortunately, our experimental results show that the proportion of these "unfavorable" portions are small (Section 6). Hence, the efficiency improvement provided by MicroFix will not be significantly degraded. In this article, we use the two voltage scheme [Usami et al. 1998] to decouple the unadaptable portions from those adaptable portions.

Two supply voltages scheme [Usami et al. 1998] can be efficiently used to handle the critical isolation in this way: apply the traditional DVFS to the minority of critical gates while complying the frequency tuning with the other portions handled by MicroFix. For more design details of the two supply voltages scheme, please refer to Usami et al. [1998].

5.2. Interpolation Clocks

Timing Interpolation is enabled by manipulating the clocks of flip-flops. Unlike the previous approaches that enable time stealing by extensively modifying the flip-flops [Joshi et al. 2007] or deploying tuning buffers [Tsai et al. 2005], we adopt a nonintrusive and buffer-free way by enabling multiple light-weighted clocks disassembled from original clock. The flip-flops with the same adaptability (have the same skew values) are driven by the same clock. Figure 7 shows a three-clock example for a circuit, where the flip-flops with the same adaptability are indicated with the same color.

We would like to point out that although multiple clocks are adopted, it is totally different from multiple clock domain designs because all clocks have the same frequency generated from the same DLL (delay-locked loop). Therefore, no multiple clock domains troubles, such as metastability and reset synchronization, are involved.

5.3. Parameter Variation Concern

The interpolation clocks can be generated with a DLL. DLLs have been widely used to reduce the clock skew across clock domains [Xanthopoulos et al. 2001; Minami et al. 2000; Jeon et al. 2004]. The detailed design of a DLL is beyond the scope of this paper. A major concern is whether those PVT (process, voltage, and temperature) variations can spoil the intentional skew. Many industry practices have shown that implementing clocks with only 10-picosecond skew is practical. For example, even in conventional tree-based clock networks across $500mm^2$ processor die with frequency up to 2.5GHz , the unintended clock skew can be efficiently limited to less than 10ps [Mahoney et al. 2005]. Moreover, we find the differences between clustered skews are usually one order of magnitude larger than the variation induced skew. Therefore, generating the interpolation clocks with well-defined intentional skew should not be a substantial problem.

In addition, the power consumption of a processor's DLLs, commonly, is less than 2% [Duarte et al. 2002], and the hardware overhead is very limited.

6. EXPERIMENTAL SETUP AND RESULTS

First, we reveal the potential of path-grained adaptability under different Timing Interpolation configurations, and then investigate a critical design tradeoff. Finally, with extensive Hspice simulations, we present the results of power reduction.

6.1. The Impact of Timing Interpolation

We first investigate the slack improvement impacted by timing interpolation for the seven largest ISCAS89 circuit benchmarks. The slack gained is measured by (slack/cycle period) $\times$ 100%. These circuits are synthesized using Synopsys Design Compiler with UMC 0.18um technology. We set the performance as the synthesizing

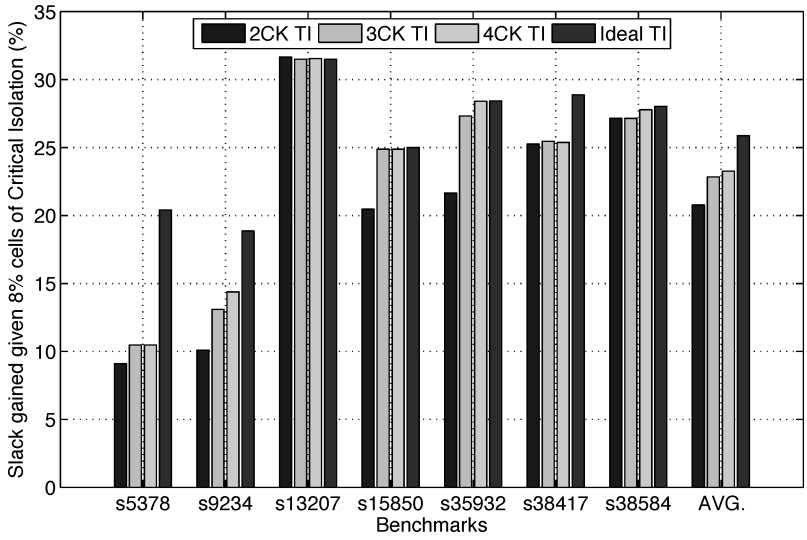


Fig. 8. Slack gained for ISCAS89 circuit benchmarks with different Timing Interpolation configurations, given 8% cells budget in Critical Isolation.

priority. Then, we analyze the path timing with PrimeTime. The cycle period is defined as the critical path delay plus 10% time margin.

In Figure 6 we have shown that three interpolation clocks can reach a close-to-ideal Timing Interpolation. In this section, we further put a bunch of results to verify this argument. Given 8% cells budget of Critical Isolation, Figure 8 shows for these benchmarks except for s5378 and s9234, 3CK configuration can yield close-to-ideal slack exploitation; more clocks, that is, 4CK, only yield very marginal improvement. Especially for S38417 and s38584, even 2CK configuration works as well as 3CK. In contrast, for s5378 and s9234, although 3CK configuration cannot fully exploit the path-grained adaptability, more clocks are still hard to accomplish this goal. Hence, 2$\sim$3CK configuration is strongly recommended.

6.2. Exploring Design Trade-Offs

We took the seven largest ISCAS89 circuit benchmarks and an industry FPU adopted by OpenSPARC T1 [Sun Microsystem Inc. 2006] as our target circuits. The power consumption (P_{total}) of a circuit can be broken down to two components: the MicroFix handled component ($P_{MicroFix}$) and the Critical Isolation component (P_{CI}). Each component is a function of voltage V and frequency F; that is,

$$P_{total} = \alpha P_{MicroFix}(V - \Delta V, F) + (1 - \alpha)P_{CI}(V, F) + P_{overhead}(V, F), \tag{16}$$

where ΔV is the reduction in voltage level for non-Critical Isolation, α is a factor determined by TH and $0 \leqslant \alpha \leqslant 1$.

Implementing MicroFix also imposes some overhead, denoted by $P_{overhead}$, which mainly comes from delay sensors and voltage converters isolating the cells in Critical Isolation. How much overhead imposed also depends heavily on the design tradeoff and will be detailed later.

On one hand, Equation (16) tells that optimizing goal should be to increase the ΔV as much as possible; on the other hand, since MicroFix cannot benefit from Critical Isolation, which implies that we should strive to shrink its scope, α should be increased

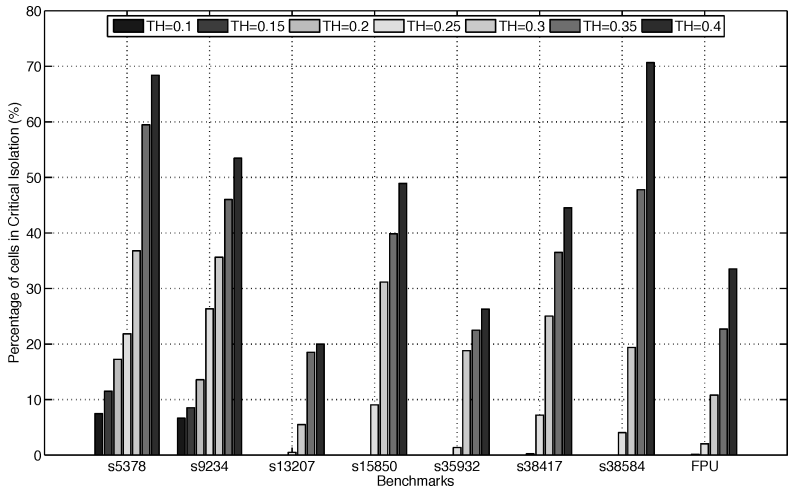


Fig. 9. Percentage of cells in Critical Path Isolation with 3CK Timing Interpolation.

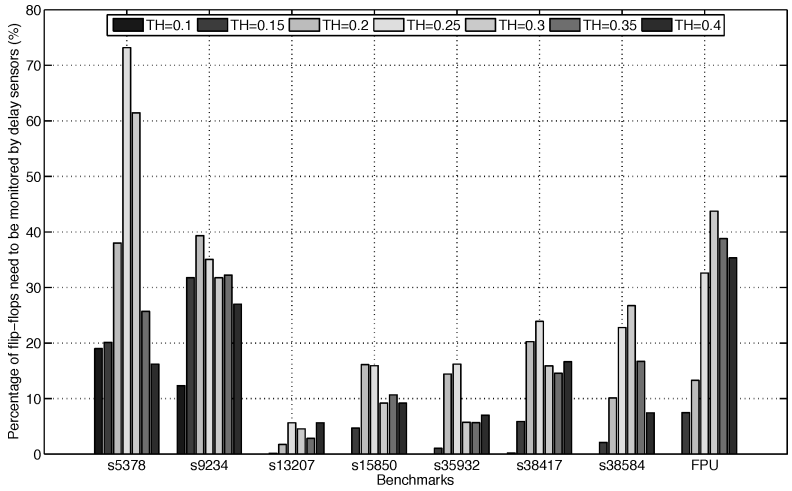


Fig. 10. Percentage of flip-flops under monitor.

as much as possible. However, the two objectives are conflict with each other, as explained in the following.

The parameter *TH* plays a critical role in the design space. Generally, larger *TH* implies more aggressive timing stealing, and thus more voltage reduction is gained; however, larger *TH* also leads to smaller proportion that MicroFix can exploit because more cells fall into Critical Isolation, which is handled by conservative voltage configurations. Figure 9 shows that the percentage of cells that fall into Critical Isolation with different *TH* expectations. Take the FPU as an example, the critical isolation is small even negligible if $TH < 0.25$, but jumps to 34% with *TH* increase from 0.25 to 0.4. The similar trend is also presented for the other circuits, though in different degrees.

6.2.1. Overhead from Delay Sensors. The sensors impose area and power overhead. The paths whose slack meet Equation (15) should be put under monitor of sensors. Figure 10 shows that the proportion of flip-flops that need to be monitored is not monotonic, but

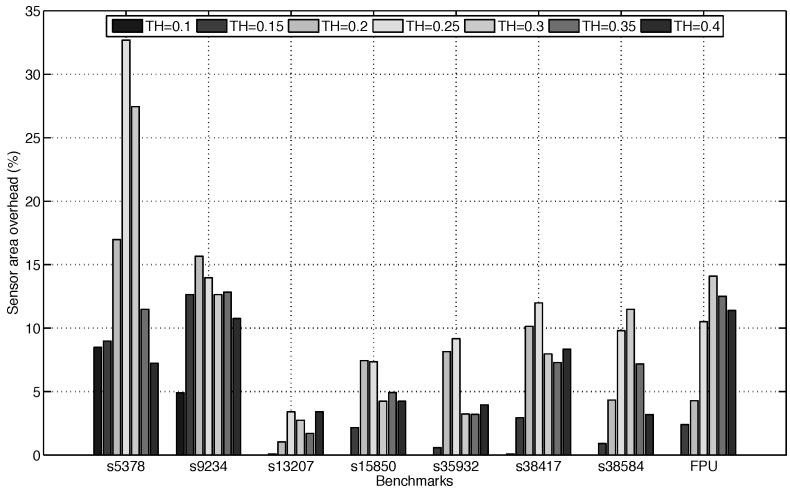


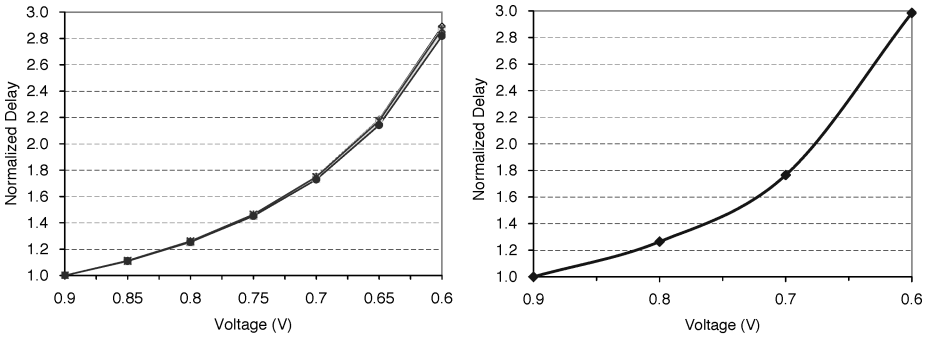
Fig. 11. Sensor area overhead.

exhibit distinct "peak" at medium *TH*. Lower *TH* resulting in less monitored flip-flops is because the population of critical path out of the Critical Isolation is still relatively small; hence fewer paths need to be put under monitor. Higher *TH* also results in fewer monitored flip-flops because more paths fall into Critical Isolation where the sensors are not required. Thus, it can be expected that medium *TH* can yield the largest percentage of critical paths out of Critical Isolation, represented as the "peak" characteristics.

Based on the results in Figure 10, we can derive the area overhead imposed by the delay sensors. The sensor configuration is as follows: each sensor is configured with eight stability checkers [Yan et al. 2009a]. The area of a sensor is about eight times that of a flip-flop. Figure 11 shows the overall area overhead for each circuit.

The sensors' power overhead is another negative impact to the MicroFix scheme. Our prior work [Yan et al. 2009a] shows that each sensor consumes eight times power of a flip-flop with per-cycle data transition in the worst case: that is each stability checker in a sensor detects a delay error simultaneously in every cycle. Take the FPU as an example, if these sensors are always on, in the most pessimistic case, the power overhead can reach up to 14% (the flip-flops' power in the FPU takes about 65% on average). Fortunately, it is not necessary to concern about such pessimistic power overhead for three reasons: 1) the sensors do not need to be always on. The sensors are turned on just in two "risky" suboperations (section 3.1). In steady states, the sensors are kept off. 2) It is impossible that every sensor captures faulty transition in every cycle. When any sensor flags a positive, the process of voltage scaling down or frequency scaling up will stop and is followed by restoring a safety margin, as Figure 5(b) illustrates. 3) The following results reveal that the configuration of $TH = 0.3$ is not an optimal choice, so such worst case actually won't happen. In brief, the sensors power overhead will not significantly degrade the efficacy of MicroFix.

6.2.2. Overhead from Critical Isolation. Dealing with the critical isolation will not come for free. The area and power overhead to isolate these cells comes from voltage level-converters. Fortunately, this power overhead is only 8% even 80% cells covered by such isolation [Usami et al. 1998]. So we can safely estimate that the level-converters imposed power overhead must be much smaller if less cells are isolated. The area overhead depends on not only the extra number of cells but also the layout constraints.



(a) Combinational delay (b) Flip-flop delay($t_{setup} + t_{c-to-q}$)

Fig. 12. Delay vs Voltage. The combinational delay comes from a bunch of ISCAS85 circuit benchmarks (the normalized delay shows very similar trend so the tag for each curve is omitted).

The accurate area overhead is unavailable yet, but based on the published data [Usami et al. 1998], a very conservative estimate is about 3% on average given $TH = 0.3$.

6.2.3. Overhead from Interpolation Clocks. Although MicroFix needs extra clocks, the clock power overhead is small because the determinant factor in clock power, clock load, dose not increase. Though some clock wires (and some buffers) overhead is imposed, which just can result in small clock power increase (typically the power of the all clock wiring takes about 7% for a clock network [Duarte et al. 2002]), the chip-level power overhead is typically less than 2.8% (given 40% clock power budget of current high performance microprocessors [Gronowski et al. 1998]).

The above experimental results have shown 1) the implication of *TH* to the scope of MicroFix (Figure 9) and 2) the overhead ($P_{overhead}$) imposed by the implementation of MicroFix. The following subsection evaluates the overall power implication with extensive Hspice simulations.

6.3. Power Reduction

The power reduction comes form the reduced voltage enabled by MicroFix. Because how much delay can be accommodated with MicroFix determines how much voltage reduction can be achieved, we first study the relation between delay and voltage, then present the power reduction.

6.3.1. Voltage Reduction. We found that although the absolute value of delay variation with voltage tuning is circuit-specific, the relative trend exhibits circuit independence, for both combinational and sequential components,[1] as Figure 12 shows. These benchmark circuits are implemented with 32nm PTM models for high-performance applications [Zhao et al. 2006], and then simulated with Hspice. The nominal voltage is 0.9V. The baseline flip-flop design is adopted by PowerPC603 processors [Gerosa et al. 1994].

With the delay-voltage relationship, we can figure out the voltage reduction under different TI configurations. The results are shown in Figure 13. At the most aggressive configuration TH = 0.4, for example, more than 15% voltage reduction can be achieved.

6.3.2. Power Comparison. The efficiency improvement represents as the reduced power consumption at given frequency, or equivalently increased frequency at a given power budget. The following discussion emphasizes the former aspects. We assume an ideal

[1]The delay increase induced by reduced voltage is reflected by two components: 1) Combinational path delay increase and 2) flip-flop delay increase. Specifically, the whole delay $T = t_c + t_{setup} + t_{c-to-q}$, where t_c is the delay of combinational critical path, t_{setup} and t_{c-to-q} is the setup time and clock-to-q time of a flip-flop.

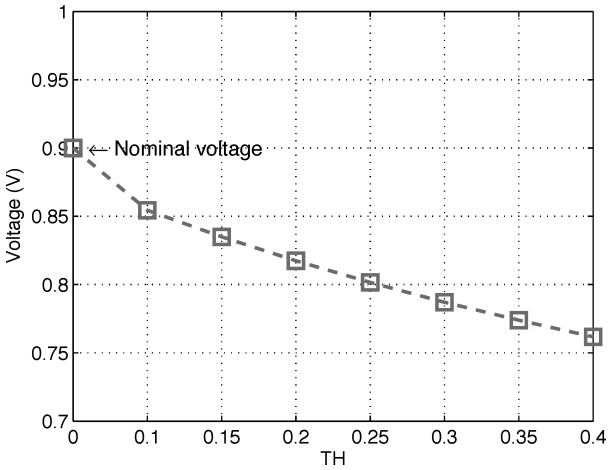


Fig. 13. MicroFix-enabled voltage for Non-Critical Isolation under different configurations.

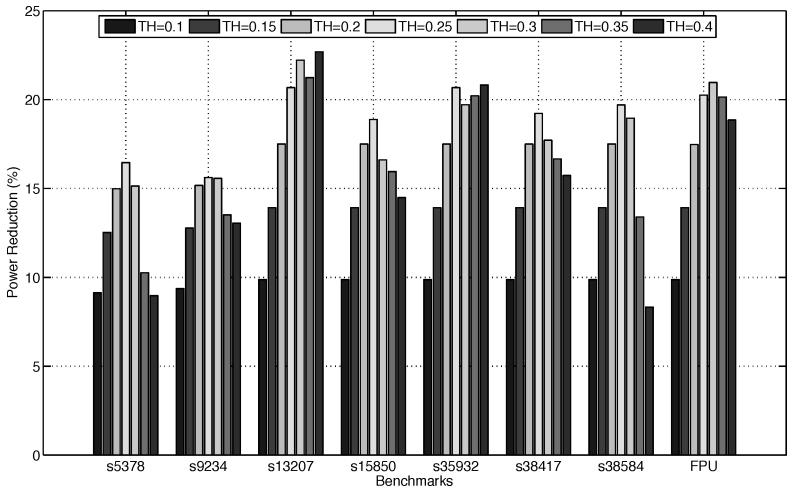


Fig. 14. MicroFix-enabled power reduction at constant frequency.

DVFS architecture as a benchmark to evaluate MicroFix architecture. Such ideal DVFS can configure the voltage and frequency without any conservativeness. Each benchmark was stressed with 1,000 random input patterns to perform power evaluation (we found 1,000 patterns are enough to exercise the circuit to reach a very stable power consumption). The timing results are based on a 180nm technology due to lack of advanced 32nm compiler libraries. To match the following analysis, we scale the STA results to the target technology based on scaling theory [Rabaey et al. 2004]).

Then, based on Equation (16), combining with the results in Figure 9 and the corresponding power overhead, we have the net power reduction, as shown in Figure 14. Generally, 20% power reduction can be achieved on average. Moreover, Figure 14 indicates that for most of the circuits, TH between 0.2 to 0.3 is an optimal choice. More aggressive *TH* will not significantly improve MicroFix's benefit, and even degrade it due to shrinking the application scope of MicroFix, as Figure 9 shows.

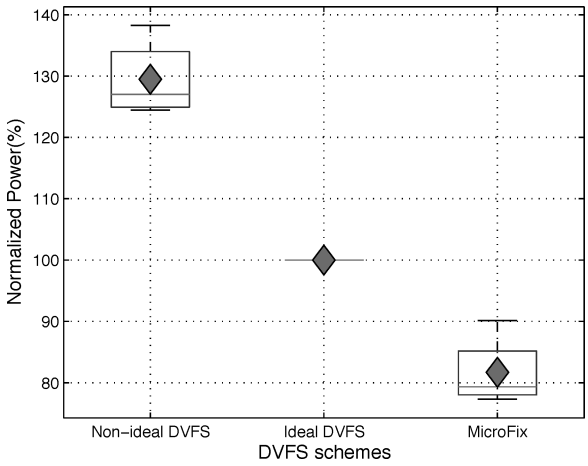


Fig. 15. Power comparison between non-ideal DVFS, ideal DVFS and MicroFix. The power for each circuit are normalized to that with the ideal-DVFS scheme. For each scheme, the average power is indicated by the dark diamond and the power variation across the set of circuits is shown with boxplot (Boxplot is a statistical illustration used to shown the median and dispersion of a group of values).

However, the potential of circuit s13207 and s35932 seems not fully exploited even with TH = 0.4, which is because the percentage of cells in Critical Isolation keeps relatively small even with TH = 0.4, as Figure 9 shows; in other words, the saved power gained from more aggressive voltage reduction can offset the loss of benefit from shrinking the scope of MicroFix. But we do not recommend more aggressive TH due to the increasing area overhead, as Figure 11 illustrates.

To further illustrate MicroFix's effectiveness, we also compare MicroFix to a nonideal DVFS scheme for which some voltage margin has to be reserved for reliable tuning operations. As a rule of thumb, 10∽20% voltage margin is common. A recent investigation for the POWER6 microprocessor, for instance, shows that the reserved voltage margin is up to 18% of the nominal supply voltage, which is 200mV for a nominal voltage of 1.1V [James et al. 2007]. In the following comparison, we assume 10% voltage margin—a relatively tight case—is reserved for the nonideal DVFS scheme. The power of each circuit is normalized to that with the ideal-DVFS scheme (no voltage margin).

The detailed results are shown in Figure 15, where not only the average power (shown with the dark diamond) is illustrated, but also the power variation (shown with boxplot) across the circuits. From these results we can see eliminating 10% voltage margin translates to 22% power reduction; furthermore, with MicroFix exploiting more fine-grained adaptability, another 16% power can be saved. It's worth noting that MicroFix can help not only eliminate voltage margin but also translate the fine-grained adaptability into power reduction; therefore, compared with nonideal DVFS, the overall power reduction offered by MicroFix is about 38% on average.

6.4. Recommended Design Choices

In the design space of MicroFix, we aim to maximize the power efficiency at the least overhead and complexity. In particular, the complexity depends heavily on 1) the interpolation clocks adopted and 2) the scope of Critical Isolation. Based on the above results, we present an empirical recommended design choice and resulting implications in Table I.

Take the s5378 for example, although the maximum power reduction can be obtained with TH = 0.25, the corresponding area overhead hovers around 33%—which

Table I. Recommended Design Choices and Resultant Implications

Circuits	TI Configuration	TH	Critical Isolation (%)	Area Overhead (%)	Power Reduction (%)
s5378	3CK	0.15	10.1	9.0	12.5
s9234	3CK	0.25	24.5	13.9	15.6
s13207	2CK	0.40	20.0	3.4	22.7
s15850	3CK	0.30	29.3	4.2	16.6
s35932	3CK	0.40	26.6	3.9	20.8
s38417	2CK	0.15	0.0	2.9	13.9
s38584	2CK	0.20	0.0	4.3	17.5
FPU	3CK	0.20	0.16	4.3	17.5

is usually unacceptable. Another alternative is adopted more aggressive TH = 0.35 for example, but 60% cells will fall into Critical Isolation at this configuration, which can significantly increase the isolation overhead. Hence, we recommend a TH = 0.15 design point, at which only 9% overhead is imposed and 10% cells need to be isolated, but the power benefit is still close to the optimum.

Table I also indicates that MicroFix is relatively unapplicable for s5378 and s9234—relatively large overhead, large isolation efforts, but modest power benefit; in contrast, the area overheads for the other circuits is no more than 5% and for some of them, that is, s38417, s38584, and FPU, almost no cell needs to be isolated at the recommended configurations. In addition, s5378 and s9234 are the two smallest circuits in our benchmarks; MicroFix tends to be more applicable for large scale circuits.

6.5. Comparison with Razor

It is hard to make an apple-to-apple comparison between MicroFix and other low-power schemes, but we give a comparative analysis between MicroFix and Razor [Ernst et al. 2003; Das et al. 2006]. Similar to Razor, MicroFix strives to avoid voltage (or frequency) conservativeness by monitoring the critical delay, though Razor achieves this goal by error detection and recovery while MicroFix achieves it by error prediction. In power comparison we conceived an conservativeness-free DVFS scheme as the comparing target which actually can be viewed as a Razor enhanced DVFS scheme.

The major difference comes from the new concept: path-grained adaptability, which opens up new opportunity for power reduction. This discovery makes MicroFix gain about 20% power reduction compared with the ideal (Razor-enhanced) DVFS scheme.

7. CONCLUSION

We proposed a novel scheme, MicroFix, to improve DVFS efficiency by exploiting the path-grained adaptability. We showed that the adaptability is abundant for most of the circuits. Then we presented Timing Interpolation technique to reap the fine-grained adaptability. An optimization model was also proposed to guide to the optimum Timing Interpolation strategy. MicroFix can also eliminate the voltage (or frequency) conservativeness by enabling the capability of delay error prediction. A set of delay sensors are deployed to indicate the minimal voltage level (or maximum frequency). We found that usually two or three interpolation clocks are competent to fully exploit the path-grained adaptability. Experimental results show that by effectively reducing the voltage margin and translating the path-grained adaptability into power benefit, MicroFix gain 38% power reduction on average, at the expense of usually less than 5% area overhead.

REFERENCES

AGARWAL, M., PAUL, B., ZHANG, M., AND MITRA, S. 2007. Circuit failure prediction and its application to transistor aging. In *Proceedings of the VLSI Test Symposium (VTS)*. 277–286.

AGGARWAL, A. AND CARLEY, L. 1999. A completely on-chip voltage regulation technique for low power digital circuits. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*.

AMD. 2000. AMD PowerNow! technology, dynamically manages power and performance. White Paper.

BORCH, E. AND TUNE, E. 2002. Loose loops sink chips. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*.

BROOKS, D., BOSE, P., SCHUSTER, S., JACOBSON, H., KUDVA, P., BUYUKTOSUNOGLU, A., WELLMAN, J., ZYUBAN, V., GUPTA, M., AND COOK, P. 2000. Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors. *IEEE Micro 20,* 6, 26–44.

DAS, S., ROBERTS, D., LEE, S., PANT, S., BLAAUW, D., AUSTIN, T., FLAUTNER, K., AND MUDGE, T. 2006. A self-tuning DVS processor using delay-error detection and correction. *J. Solid-States Circuits*. *41,* 4, 792–804.

DORSEY, J., SEARLES, S., CIRAULA, M., JOHNSON, S., BUJANOS, N., WU, D., BRAGANZA, M., MEYERS, S., FANG, E., AND KUMAR, R. 2007. An integrated quad-core opteron processor. In *Proceedings of the International Solid-States Circuits Conference (ISSCC)*. 102–103.

DRAGONE, N., AGGARWAL, A., AND CARLEY, L. R. 2000. An adaptive on-chip voltage regulation technique for low-power applications. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 20–24.

DUARTE, D. E., VIJAYKRISHNAN, N., AND IRWIN, M. J. 2002. A clock power model to evaluate impact of architectural and technology optimizations. *IEEE Trans. VLSI*, 844–855.

ELKAN, C. 2003. Using the triangle inequality to accelerate K-means. In *Proceedings of the International Conference on Machine Learning (ICML)*.

ERNST, D., KIM, N., DAS, S., PANT, S., RAO, R., PHAM, T., ZIESLER, C., BLAAUW, D., AUSTIN, T., FLAUTNER, K., AND MUDGE, T. 2003. Razor: A low-power pipeline based on circuit-level timing speculation. *Micro*, 7–18.

GEROSA ET AL. 1994. A 2.2 W, 80 MHz superscalar RISC microprocessor. *J. Solid-States Circuits*. *29,* 12, 1440–1452.

GRONOWSKI, P. E., BOWHILL, W., PRESTON, R., GOWAN, M., AND ALLMON, R. 1998. High performance microprocessor design. *J. Solid-States Circuits*. *33,* 5, 676–686.

HERBERT, S. AND MARCULESCU, D. 2007. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 38–43.

INTEL. 2004. Enhanced Intel SpeedStep technology for the Intel Pentium M processor. White Paper.

ISCI, C., BUYUKTOSUNOGLU, A., CHER, C., BOSE, P., AND MARTONOSI, M. 2006. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. *Micro*, 347–358.

JAMES, N., RESTLE, P., FRIEDRICH, J., HUOTT, B., AND MCCREDIE, B. 2007. Comparison of split-versus connected-core supplies in the POWER6 microprocessor. In *Proceedings of the International Solid-States Circuits Conference (ISSCC)*. 298–299.

JEON, Y., LEE, J., LEE, H., JIN, K., MIN, K., CHUNG, J., AND PARK, H. J. 2004. A 66-333-MHz 12-mW register-controlled DLL with a single delay line and adaptive-duty-cycle clock dividers for production DDR SDRAMs. *J. Solid-States Circuits*. 2087–2092.

JOSHI, V., BLAAUW, D., AND SYLVESTER, D. 2007. Soft-edge flip-flops for improved timing yield: Design and optimization. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 667–673.

MAHONEY, P., FETZER, E., DOYLE, B., AND NAFFZIGER, S. 2005. Clock distribution on a dual-core, multi-threaded itanium-family processor. In *Proceedings of the International Solid-States Circuits Conference (ISSCC)*. 292–293, 599.

MCGOWEN, R., POIRIER, C., BOSTAK, C., IGNOWSKI, J., MILLICAN, M., PARKS, W., AND NAFFZIGER, S. 2006. Power and temperature control on a 90-nm Itanium family processor. *J. Solid-States Circuits*. *41,* 1, 229–237.

MINAMI, K., MIZUNO, M., YAMAGUCHI, H., NAKANO, T., MATSUSHIMA, Y., SUMI, Y., SATO, T., YAMASHIDA, H., AND YAMASHINA, M. 2000. A 1 GHz portable digital delay-locked loop with infinite phase capture ranges. In *Proceedings of the International Solid-States Circuits Conference (ISSCC)*. 350–351, 469.

RABAEY, J., A.CHANDRAKASAN, AND B.NIKOLIC. 2004. *Digital Integrated Circuits, A Design Perspective*. 2nd Ed. Pearson Education Asia Limited and Tsinghu University Press, Chapter 9.

SAKURAI, T. AND NEWTON, A. 1990. Alpha-power law MOSFET model and its application to CMOS inveter delay and other formulas. *J. Solid-States Circuits*. *25*, 584–594.

SUN MICROSYSTEM INC. 2006. OpenSPARC T1 microarchitecture specification.

TIWARI, A., SARANGI, S. R., AND TORRELLAS, J. 2007. ReCycle: Pipeline adaptation to tolerate process variation. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*. 323–334.

TSAI, J., ZHANG, L., AND CHEN, C. 2005. Statistical timing analysis driven post-silicon-tunable clock-tree synthesis. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 574–580.

USAMI, K., IGARASHI, M., MINAMI, F., ISHIKAWA, T., KANZAWA, M., ICHIDA, M., AND NOGAMI, K. 1998. Automated low-power technique exploiting multiple supply voltages applied to a media processor. *J. Solid-States Circuits*. *33,* 3, 463–472.

XANTHOPOULOS, T., BAILEY, D. W., GANGWAR, A. K., GOWAN, M. K., JAIN, A. K., AND PREWITT, B. K. 2001. The design and analysis of the clock distribution network for a 1.2 GHz Alpha microprocessor. In *Proceedings of the International Solid-States Circuits Conference (ISSCC)*. 402–403.

YAN, G., HAN, Y., AND LI, X. 2009a. A unified online fault detection scheme via checking of stability violation. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*. 496–501.

YAN, G., HAN, Y., LIU, H., LIANG, X., AND LI, X. 2009b. MicroFix: Exploiting path-grained timing adaptability for improving power-performance efficiency. In *Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED)*. 395–400.

ZHAO, W. AND CAO, Y. 2006. New generation of Predictive Technology model for sub-45nm early design exploration. *IEEE Trans. on Electron Devices 53,* 11, 2816–2823.

Received March 2010; revised December 2010; accepted December 2010