MicroFix: Exploiting Path-grained Timing Adaptability for Improving Power-Performance Efficiency

Guihai Yan
1.Key Laboratory of Computer System and Architecture,
ICT, CAS, Beijing, P.R. China
2.Graduate University of CAS
yan_guihai@ict.ac.cn

Yinhe Han
Key Laboratory of Computer System and Architecture,
ICT, CAS, Beijing, P.R. China
yinhes@ict.ac.cn

Hui Liu
Key Laboratory of Computer System and Architecture,
ICT, CAS, Beijing, P.R. China
liuhui@ict.ac.cn

Xiaoyao Liang
NVIDIA Corporation
USA
xliang@nvidia.com

Xiaowei Li*
Key Laboratory of Computer System and Architecture,
ICT, CAS, Beijing, P.R. China
lxw@ict.ac.cn

ABSTRACT

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 approach, MicroFix, to exploit such fine-grained adaptability. We first reveal the potential of the path-grained timing imbalance and then present a novel implementation of MicroFix. Moreover, to eliminate the conservative margins of traditional DVFS, unlike the previous approaches that reactively handle the delay errors (induced by aggressively scaled voltage/frequcncy) by error detection and recovery strategies, we propose a proactive approach by error prediction. MicroFix was evaluated based on the floating-point unit adopted by OpenSPARC T1 processor. Compared against traditional DVFS schemes, the experimental results shows that MicroFix improves the EDP (Energy-Delay Product) up to 35% for high-performance circuits and PDP (Power-Delay Product) to 28% for low-power circuits, while at the expense of only 7% area overhead.

Categories and Subject Descriptors

B.7 [**Integrated Circuits**]: General

General Terms

Design, Performance, Reliability

Keywords

Timing adaptability, DVFS, efficiency

1. INTRODUCTION

The object of dynamic voltage/frequency scaling (DVFS) [14] [18] is to achieve an optimal efficiency, i.e. power-performance tradeoff, though the efficiency may imply different preference between power and performance [4]. In essence the bottleneck of scaling down voltage is the potential delay errors on some critical paths. Traditional DVFS approaches eliminate such errors by reducing the frequency of corresponding processor cores [3][9]. Such coarse-grained approaches are naturally oblivious to fine-grained, i.e. path-grained, adaptability that can be exploited to more pertinently accommodate the potential delay errors caused by reduced voltage or increased frequency. We propose a path-grained adaptation scheme, MicroFix (short for MicroFix-enhanced DVFS), to achieve a better power-performance tradeoff.

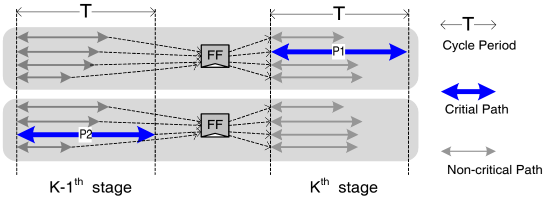


Figure 1: Impact of DVFS to Path Delay

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 error if no corresponding frequency scaling involved at this time. This fact is derived from a coarse-grained perspective. However, from a path-grained perspective such potential delay error can be eliminated without sacrificing the frequency as long as intentional time stealing is conducted, because for $P1$ all of the related upstream paths are timing non-critical, and for $P2$ all of the downstream paths are also timing non-critical. Actually, such path-timing imbalance, called intrinsic imbalance, are very common in pipelines (Section 2). Furthermore, we find that given a constant frequency, voltage scaling down can exacerbate the path-grained timing imbalance. MicroFix aims to exploit such path-grained timing imbalance to minimize the performance penalty, thereby achieving a better power-performance tradeoff.

The proposed path-grained timing adaptation in essence is timing stealing. Traditionally, the timing adaptation for a pipeline is conducted at coarse-grained, i.e. stage-grained. ReCycle [2] 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.

*To whom correspondence should be addressed.

ISLPED'09, August 19–21, 2009, San Francisco, California, USA.
Copyright 2009 ACM 978-1-60558-684-7/09/08 ...$5.00.

One obstacle of MicroFix arises from this case: a flip-flop is sandwiched between critical paths, called *unadaptable flip-flop*. Such flip-flops actually serves as the decisive factor determining the efficiency of traditional DVFS approaches. Fortunately, our experimental results shows that the proportion of these "unfavorable" flip-flops is small (section 2). Two supply voltages scheme [12] can be efficiently used to handle such case: apply the traditional DVFS to the minority of gates (a subset of critical path isolation [17]) that lead to unadaptable flip-flops while complying the frequency tuning with the other portions handled by MicroFix. In this paper, we omit the design details of two supply voltages scheme[12].

We have evaluated MicroFix based on a gate-level commercial floating point unit (FPU), using two types of 32nm PTM models [22] dedicated to high performance and low power applications, respectively. The experimental results show that MicroFix can improve the high performance efficiency EDP by 35%, low-power efficiency PDP by 28%, at the expense of 7% area overhead.

The next section reveals the timing imbalance of the FPU and describes the effect of MicroFix on DVFS. Section 3 presents the implementation. Section 4 evaluates MicroFix, followed by conclusion in Section 5.

2. PATH-GRAINED TIMING IMBALANCE

We focus on the imbalance that can be exploited for improving DVFS efficiency. Such timing imbalance can be reflected by characterizing pipeline flip-flops. A pipeline 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 $\{s_1, s_2, \ldots, s_n\} > TH$.
- Forward Adaptable Flip-flop (FAFF): All of the $\{e_1, e_2, \ldots, e_m\} > TH$, but at least one of the $\{s_1, s_2, \ldots, s_n\} \leq TH$.

  FAFFs and BAFFs can also serve as margin provider, although in a "unidirectional" manner.
- Unadaptable Flip-flop (UAFF): At least one of the $\{e_1, e_2, \ldots, e_m\} \leq TH$, and at least one of the $\{s_1, s_2, \ldots, s_n\} \leq TH$.

Threshold TH is a key factor affecting the design tradeoffs. On one hand, the larger TH, the higher 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 non-critical portion. Section 4 will discuss the detailed implication of TH.

Note that a path may start and end with the same flip-flop, i.e. a path in a tight loop [6], but the above rules are still valid.

The following discusses the path-grained timing imbalance from two aspects: 1) intrinsic imbalance and 2) how does DVFS affect such imbalance.

2.1 Intrinsic Imbalance

As a case study, we choose a pipelined FPU adopted by OpenSPARC T1 [19] 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).

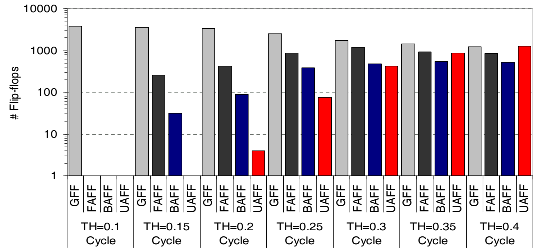


Figure 2: Distribution of Flip-flops with Different TH

Figure 2 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. This margin can be exploited with the previous DVS approaches [15] without sacrifice frequency (so not "DVFS"). 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 time stealing; thereby achieving the object: trading less frequency degradation with the same voltage reduction, or equivalently, obtaining more voltage reduction with the same frequency degradation.

2.2 DVFS Exacerbating Imbalance

Reducing voltage results in different delay change for different paths. Generally, the longer paths suffer more delay increase than short ones. 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 $m < n$. 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.

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 $(1/T)$ to F' $(1/T')$ then followed by corresponding voltage scaling.

Before the voltage scaling, the delay of P_a and P_b can be expressed as

$$t_{P_a} = m \times t_G \text{ and } t_{P_b} = n \times t_G.$$

The corresponding slack of P_a is $S_{P_a} = T' - t_{P_a}$. Similarly, $S_{P_b} = T' - t_{P_b}$.

After the voltage scaling, each gate suffers delay increase by δ (δ is a function of voltage), then we get

$$t'_{P_a} = m \times (t_G + \delta) \text{ and } t'_{P_b} = n \times (t_G + \delta).$$

The slack of P_a is $S'_{P_a} = T' - t'_{P_a}$; similarly, $S'_{P_b} = T' - t'_{P_b}$.

The difference between S_{P_a} and S_{P_b}, denoted as $\Delta S = S_{P_a} - S_{P_b}$, can serve as a metric to evaluate the degree of timing imbalance between P_a and P_b. Then,

$$\Delta S = (T' - m \times t_G) - (T' - n \times t_G) = (n - m) \times t_G;$$

$$\Delta S' = (T' - m \times (t_G + \delta)) - ((T' - n \times (t_G + \delta)) = (n - m) \times (t_G + \delta).$$

Clearly, $\Delta S < \Delta S'$, which implies this step of voltage scaling results in more timing imbalance between P_a and P_b. This timing imbalance can be also stolen to compensate a fraction of frequency degradation.

2.3 Utilizing Imbalance

From traditional DVFS perspective, the lower bound of T' is

$$T'_1 = n(t_G + \delta), \tag{1}$$

while from MicroFix's perspective, the lower bound of T' is

$$T'_2 = \frac{m + n}{2}(t_G + \delta). \tag{2}$$

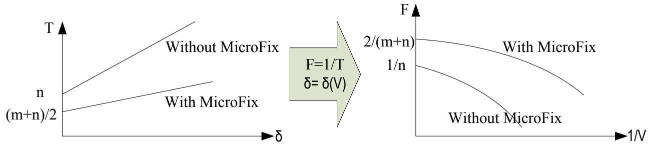


Figure 3: Effect of MicroFix on DVFS

Because $T_2' < T_1'$, less performance degradation is imposed. Note that in the fist order model $m(t_G + \delta)$ and $n(t_G + \delta)$ can be viewed as the delay of the two paths. Based on Eq.(1) and (2), we present the effect of MicroFix on DVFS in Figure 3. The δ non-linearly dependents on voltage V which can be obtained from evaluation results (Section 4). Generally, MicroFix can slow down the degradation of frequency in DVFS operations. The next section presents the implementation of MicroFix.

3. IMPLEMENTATION OF MICROFIX

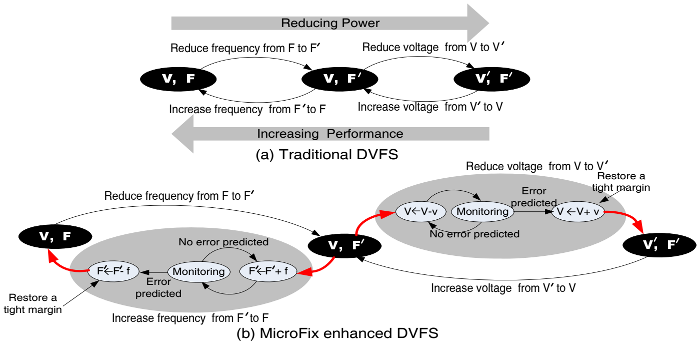
Figure 4 shows MicroFix scheme. The main differences from the traditional DVFS scheme come from three aspects: 1) two voltage supplies are deployed: the conservative one, which provide the voltage supply for the critical isolation (Section 2), is controlled by traditional DVFS, and the normal one, which for the other portion outside of the critical isolation, is controlled by MicroFix; 2) timing sensors are employed to enable a self-tuning DVFS; 3) two extra clocks are used to realize intentional time stealing. First we present the operational principle of MicroFix, and then explain two implementation issues.

3.1 Operational Principle

We use timing 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 [14]. Such sensors trigger the DVFS units to tune voltage-frequency to a pre-defined level. Although the off-line determined voltage-frequency configurations are effective, they have been proved to be quite conservative [15] since these configurations do not directly target delay errors which serve as a more essential factor than temperature or power to DVFS operations. Based on this fact, Razor scheme equips a pipeline with delay error detection capability and takes the error detection result as an indicator to voltage reduction to eliminate the conservative voltage margin, thereby obtaining a more aggressive power reduction [15].

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 — timing sensor — are as cost-efficient as detection logics of Razor. Note that timing sensors are not aimed to replace the temperature and power sensors, but to improve the traditional DVFS schemes. A brief introduction of a timing sensor is presented in the next subsection.

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 sub-operations that can potentially induce delay errors: 1) reducing voltage and 2) increasing frequency. The modified sub-operations are shown in Figure 5(b). First, the timing sensors are turned on just before conducting the "risky" sub-operations; second, the voltage/frequency control unit progressively reduces the voltage (or increase frequency) under the monitoring of timing 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 timing sensors off.


Reducing Power
Reduce frequency from F to F′
Reduce voltage from V to V′
V, F
V, F′
V′, F′
Increase frequency from F′ to F
Increase voltage from V′ to V
Increasing Performance
Reduce voltage from V to V′
Restore a tight margin
Reduce frequency from F to F′
Error predicted
V←V-v
Monitoring
V ←V+ v
No error predicted
V, F
No error predicted
V, F′
V′, F′
F′←F′- f
Monitoring
F′←F′+ f
Error predicted
Restore a tight margin
Increase voltage from V′ to V
Increase frequency from F′ to F

(a) Traditional DVFS

(b) MicroFix enhanced DVFS

Figure 5: MicroFix enhanced DVFS

We would like to note that the safety during the two "risk" operations is guaranteed by a set of properly deployed sensors: all of the BAFFs, FAFFs (except that involved in critical isolation), and even a subset of GFFs are monitored by timing sensors (Section 4.1). Thus, any of the sensors flags an alarm, the corresponding adjustment is immediately stopped and followed by restoring a safe margin.

The v (or f) depends on the width of guard band [1] of timing sensors. The following presents several quantitative DVFS parameter relations.

Suppose the resolution of voltage scaling is R_V and frequency resolution is R_F. Then

$$\delta_V = f(V, R_V) \text{ and } \delta_F = g(F, R_F) \tag{3}$$

where δ_V and δ_F denotes the delay change caused by one step of (atomic) voltage and frequency scaling. Assume $R_V \ll V$ and $R_F \ll F$, then $\delta_F \approx (1/F)^2 \times R_F$ because $F = 1/T$. 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. Then $\delta_V \approx (\mathrm{d}f_{V \to D}/\mathrm{d}V) \times R_V$.

To avoid the delay change induced by one step voltage or frequency turning runs out of the prediction capability, R_V, R_F, and the sensor's guard band T_{GB} [1] (a specified period of time near the end of a cycle when the monitored signals should have been stabilized) should be calibrated to meet the following relation:

$$\delta_V < T_{GB} \text{ and } \delta_F < T_{GB} \tag{4}$$

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

$$T_{GB} = f(V', v), \text{ and } T_{GB} = g(F, f) \tag{5}$$

to safely accommodate the predicted delay errors.

In fact, Eq.(4) can be easily satisfied. Generally, for a 15-20 Fo4 pipeline fabricated using 45nm technology, 10mV of R_V, which can actually be hundreds of microvolt magnitude [14], results in one-step delay change about 20ps. R_F only required at 1-MHz to achieve picosecond magnitude of delay change for a 1-GHz pipeline. While MicroFix adopting the T_{GB} setting of 5% of clock period, which results in 50ps guard band, is very practical [1]. Note that because the cycle period T is not constant, so is T_{GB} and related v and f.

As a tradeoff, the tuning latency is increased. Considering the promising Razor [15] also encounters the similar situation, we believe such latency won't substantially hurt the effectiveness of MicroFix.

3.2 Implementation Issues

3.2.1 Clock Assignment and Generation

The intentional time stealing is enabled by manipulating the clock of pipeline flip-flops. Unlike the previous approaches that enable time stealing by extensively modifying the flip-flops such as [21], we adopt a non-intrusive way by introducing two extra clocks. MicroFix does not need to individually manipulate the clock driving each flip-flops. The flip-flops with the same adaptability are driven by the same clock. Specifically, all FAFFs are clocked by a forward skewed clock (FCLK) to enable forward time stealing; all BAFF are clocked by a backward skewed clock (BCLK) to enable backward time stealing. The other flip-flops are clocked by original clock (CLK) because they either do not

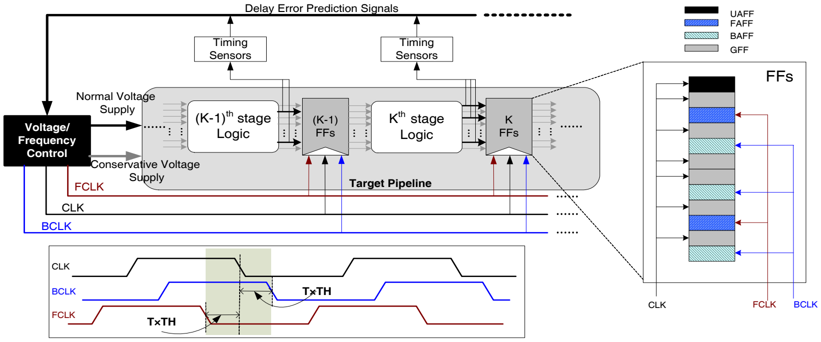


Figure 4: Top view of MicroFix scheme

need to be adjusted (GFFs) or have no timing margin can be used (UAFFs). Figure 4 also shows an example of clock assignment for a set of pipeline flip-flops.

Clock FCLK and BCLK, with intentional skew ($T \times TH$) from CLK, can be generated with a DLL (delay-locked loop). DLLs have been widely used to reduce the clock skew across clock domains [20][11][23]. The detailed design of a DLL is beyond the scope of this paper. The 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 picoseconds of skew is very 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 less than 10ps [13]. Thus, it can be extrapolated that for relatively spatial concentrated pipeline logics with less die area, the unintentional skew can be further optimized. In fact, even "10ps" is generally one order of magnitude smaller than the intentional skew. Moreover, the power consumption of a processor's DLLs, commonly, is less than 2% [5], and the hardware overhead is very limited. Therefore, generating the extra clocks with intentional skew should not be a substantial problem.

We would like to point out that although MicroFix needs two extra clocks, the clock power overhead is small because the determinate factor in clock power—clock load—increases little. Though some clock wiring 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[5]), the chip-level overhead is almost negligible.

3.2.2 *Timing sensors*

Two types of sensors can be employed. The first type is based on voltage sampling and comparison [16], and the second is based on signal stability checking [1][8]. The first type needs some analog logics and the second involves dynamic logics. Both types are overhead (power and area) efficient. In terms of sensor deployment, the second types are more trivial than the first. In this paper, we use the second type. The following gives a brief description of this type of sensors. The details can be obtained from [1][8].

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 [8]. Agarwal et al. used such stability violation for predicting aging induced delay errors [1]. 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 [8].

4. EXPERIMENTAL SETUP AND RESULTS

MicroFix has been evaluated from two aspects. First, we reveal the potential of path-grained adaptability and investigate the tradeoff between adaptability and overhead based on a gate-level FPU. Then, with extensive Hspice simulations, we present the quantitative efficiency improvement for two versions of design: high-performance and low-power version based on two types of 32nm PTM CMOS models [22].

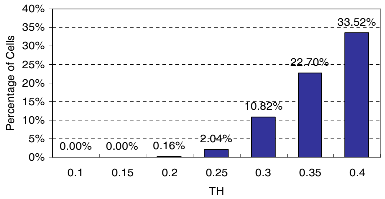


Figure 6: Percentage of Cells in Critical Path Isolation

We use the FPU adopted by OpenSPARC T1 [19] as our target pipeline. Each sensor is configured with eight stability checkers. The area of a sensor is about eight times that of a pipeline flip-flop (based on the number of transistors). In the worst case, that is every stability checker in a sensor detects a delay error in every cycle, the power of a sensor is close to that of eight flip-flops with data transition rate of 1 [8] (given a period of time, data transition rate is defined as the ratio of the number of transitions at a flip-flop's data output q to the number of cycles).

The intrinsic path-grained timing imbalance has been presented in Section 2. The next subsection discusses how does TH affect the design tradeoffs.

4.1 Exploring Design Tradeoffs

TH affects the design from two aspects: 1) the proportion of target pipeline that MicroFix applied to and 2) the number of sensors required.

Generally, larger TH implies more aggressive timing stealing can be realized; however, the smaller proportion that MicroFix can exploit because more cells leading to UAFFs fall into "critical path isolation" which is handled by conservative DVFS configurations. Note that, in MicroFix, **not all critical paths are included by such isolation**, which significantly differs from multiple supply voltage scheme [12].

Figure 6 shows that the percentage of cells that fall into critical path isolation is small even negligible if $TH < 0.25$, but jumps to 34% with TH increase from 0.25 to 0.4. 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 [12]. So we can safely estimate that the level-converters imposed power overhead must be less than 3% with TH=0.3. The area overhead depends on not only the extra number of cells but also the layout constraints. The accurate area overhead is unavailable yet, but based on the published data [12], a very conservative estimate is 3% at TH=0.3.

The sensors also impose area and power overhead. The paths in the critical path isolation and those paths with slack less than

$$T \times TH + t_{margin}$$

(commonly, $t_{margin} = T \times 10\%$) do not need to be monitored

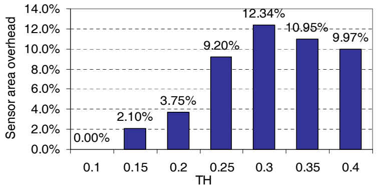


Figure 7: Sensor Area Overhead

Table 1: Summary of ISCAS85 benchmarks adopted

Circuit	Function	No. Inputs	No. Outputs	No. Logic Gates
c432	27-channel interrupt controller	36	7	160
c499	32-bit SEC circuit	41	32	202
c880	8-bit ALU	60	26	383
c1355	32-bit SEC circuit	41	32	546
c1908	16-bit SEC/DED circuit	33	25	880
c2670	12-bit ALU and controller	233	140	1193

by timing sensors. Thus, only the BAFFs, FAFFs, and a subset of GFFs need to be monitored. Figure 7 indicates the peak overhead is about 12% at TH=0.3. Larger TH results less sensor overhead because more paths fall into critical isolation where the sensors are not required. This observation again implies that overly aggressive time stealing does not necessarily results in more efficient MicroFix.

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, and a sensor takes the eight times the power of a flip-flop). However, such pessimistic power overhead won't offset much efficiency of MicroFix for three reasons: 1) the sensors do not need to be always on. The sensors are turned on just in two "risky" sub-operations (section 3.1). In steady states, the sensors are kept off. 2) It is impossible that every sensor capture 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.

The following subsection presents the experimental results with extensive Hspice simulations.

4.2 Experimental Results

The qualitative implications of TH can be studied with a gate-level version of FPU, but the detailed delay and power relation of the target pipeline is hard to obtain at this level because the FPU is too large to be simulated with Hspice. We preset an indirect method to evaluate the effects of MicroFix. The following justifies the evaluation method and discusses the results.

The total power P_{total} consists of two parts: combinational logic's power and pipeline flip-flops' power. Each part is a function of voltage V and frequency F; that is

$$P_{total}(V,F) = P_{comb}(V,F) + P_{ff}(V,F)$$

where, $1/F = T = t_c + t_{setup} + t_{c-to-q}$; 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 (we omit the trivial margin term).

4.2.1 Combinational Component

We use a subset of ISCAS85 circuit benchmarks to evaluate the combinational component of the target pipeline. Using these benchmarks to evaluate the target pipeline is still effective because the key observation: the normalized power-voltage (P-V) and delay-voltage (D-V) relations are independent of the scale of the circuits. Table 1 shows the adopted circuit benchmarks with scale (number of gates) across about one order of magnitude. All of the circuits were implemented with 32nm PTM models for high-performance (HP) and low-power (LP) applications [22], respectively, and then simulated with Hspice. The nominal voltage is 0.9v for HP models and 1.0v for LP models.

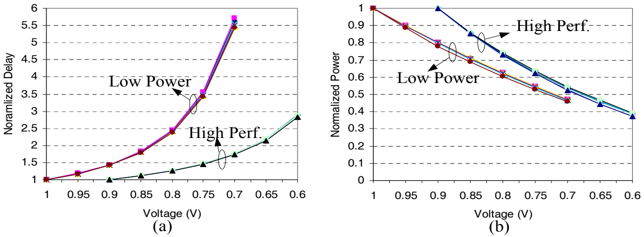


Figure 8: Impact of Voltage Scaling on Combinational logics (Note that power is calculated at constant frequency determined at lowest voltage level)

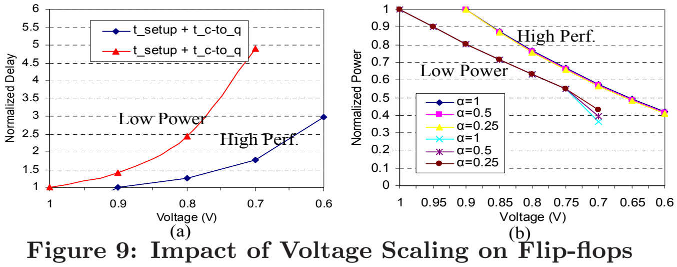


Figure 9: Impact of Voltage Scaling on Flip-flops

Each benchmark was stressed with 1,000 random input patterns to perform power evaluation (we found 1,000 patterns, which takes about 40-hour simulation time for c2670, are enough to exercise the circuit to reach a very stable power consumption). Moreover, the accurate relation between path delay and voltage is necessary. We use delay-test pattern to sensitize the critical path for delay evaluation. The steps in delay evaluation are described as following:

- Step 1: Identify the most critical path of the synthesized verilog version of the bunch of benchmarks with PrimeTime.
- Step 2: Use TetraMAX to generate delay-test pattern targeting the selected critical path.
- Step 3: Measure the delay of the path with Hspice.

These patterns are translated into PWL (piecewise linear) voltage sources to serve as the stimuli of the Hspice version benchmarks. Each stimulus is reshaped with a pair of inverters to obtain a close-to-realistic transition slew rate.

Figure 8 shows the normalized P-V and D-V curve of the HP and LP version of benchmarks. Two facts can be derived from the results: 1) the normalized P-V and D-V relation is almost independent of the scale and function of circuits, so we extrapolate that the P-V and D-V relations for the combinational component of target FPU still comply with the above results. 2) The change of power for HP version is more sensitive to voltage scaling than that of LP version; in contrast, the change in delay for HP version is much less sensitive to voltage scaling than that of LP version. This difference motivates us to conduct two version of evaluations.

4.2.2 Sequential Component

We take a flip-flop design adopted by PowerPC603 processors [7] as the baseline. Besides voltage, given a constant frequency, the power of a flip-flop also depends on the transition rate of data, denoted as α. The delay of a flip-flop is the sum of setup time and clock-to-q time.

Figure 9 indicates that the D-V and P-V is very close to that of combinational logics. Moreover, the normalized P-V is independent to α. In addition, for P-V with LP model, there seems an abrupt drop when voltage decreases from 0.75v to 0.7v; this is because the excessively low level of voltage causes the flip-flop fail to capture and hold the data and thus should be avoided.

Based on Figure 8 and 9, we conclude that the voltage scaling have very close impact on the combinational logics and flip-flops of the target pipeline; thus treating them equally is reasonable.

The following presents the efficiency comparison of MicroFix against traditional ideal DVFS—each voltage-frequency is configured without "coarse-grained" margins. Even so, MicroFix still achieves significant efficiency improvement, as following explains.

4.2.3 *Efficiency Comparison*

For HP version of pipeline, the efficiency is defined as EDP (energy-delay product), and for LP version defined as PDP (power-delay product) [4]. Based on the above results (Figure 8 and 9), Figure 10 presents the EDP and PDP trends, where "without MicroFix" means that the DVFS just complies with the delay-power (voltage) relation of the most critical paths—that is actually the efficiency upper bound of traditional DVFS, while "with MicroFix" denotes that the critical paths that are out of the critical isolation can obtain extra margins by enable corresponding path-grained adaptation. Figure 10(a) and (b) shows the P-D relations under different DVFS schemes. These results indicate that, at the beginning of voltage scaling down, trading performance (delay) for power is highly effective for both HP and LP version of circuits. But as the circuits get into low-power period, such tradeoff will be much less effective because the change in power is less sensitive to that in delay. This trend will result in considerable degradation in efficiency. Figure 10(c) and (d) confirm this conclusion.

Then we apply these results to the target FPU, and take account of the impacts of critical isolation and sensor's overhead (the STA results actually 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 [10]). We conclude that TH=0.2 is a optimal choice — 35% and 28% efficiency improvement are achieved respectively before the power is reduced to 50% level, at the expense of only 7% area overhead (4% from sensors and 3% from level converters) and negligible isolation efforts (only less 2% cells need to be isolated). More aggressive TH won't significantly improve MicroFix's performance, and even degrade it due to shrinking the application scope of MicroFix. For example, with TH increasing from 0.2 to 0.3, the maximum efficiency improvement is only about 5%, but the area overhead increase up to 10%; while TH increases from 0.3 to 0.4, the efficiency is not improved but decreased to close the level of TH=0.2. While the area overhead is more than 10%, and more cells need be isolated thereby resulting more design complexity.

5. CONCLUSION

We have proposed a novel scheme, MicroFix, to improve DVFS efficiency by exploiting the path-grained adaptability. Such adaptability comes from the commonly existed path delay imbalance. Furthermore, we reveal that the traditional DVFS can also aggravate such imbalance and present an efficient implementation of MicroFix by employing timing sensors. Evaluation results shows that excessively exploiting the path-grained imbalance will not necessarily lead to higher efficiency. Compared against traditional DVFS schemes, MicroFix can improve the efficiency of EDP for high-performance application up to 35% and PDP for low-power application up to 28% in the optimum case, at the expense of only 7% area overhead.

6. ACKNOWLEDGMENTS

This work was supported in part by the High-Tech Research and Development Program of China under grant No.(2007AA01Z-109, 2008AA01Z147) and National Natural Science Foundation of China under grant No.(60806014, 60633060, 60831160526, 60776031), and by the National Basic Research Program of China under grant No.(2005CB321604, 2005CB321605).

7. REFERENCES

[1] M. Agarwal,, B.C. Paul, M. Zhang, and S. Mitra. Circuit Failure Prediction and Its Application to Transistor Aging. *VTS*, pages 277–286, 2007.

[2] A. Tiwari, S. R. Sarangi, and J. Torrellas. ReCycle: Pipeline Adaptation to Tolerate Process Variation. *ISCA*, pages 323–334, 2007.

[3] C. Isci, A. Buyuktosunoglu, C. Cher, P. Bose, and M. Martonosi. An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget. *Micro*, pages 347–358, 2006.

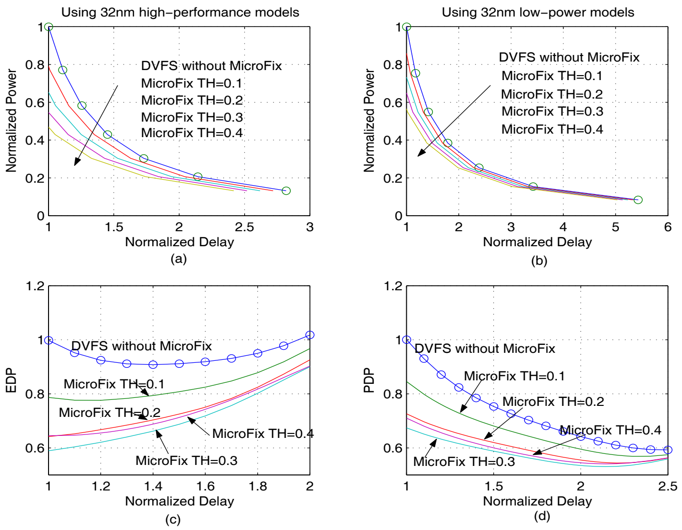


Figure 10: EDP and PDP Comparisons

[4] D. Brooks, P. Bose, S. Schuster, H. Jacobson, P. Kudva, A. Buyuktosunoglu, J. Wellman, V. Zyuban, M. Gupta and P. Cook. Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors. *IEEE Micro*, 20(6):26–44, Nov.-Dec. 2000.

[5] D. E. Duarte, N. Vijaykrishnan, and M. J. Irwin. A clock power model to evaluate impact of architectural and technology optimizations. *IEEE Trans. on VLSI*, pages 844–855, 2002.

[6] E. Borch, and E. Tune. Loose Loops Sink Chips. *HPCA*, 2002.

[7] G. Gerosa et al. A 2.2 W, 80 MHz superscalar RISC microprocessor. *JSSC*, 29(12):1440–1452, Dec 1994.

[8] G. Yan, Y. Han, and X. Li. A Unified Online Fault Detection Scheme via Checking of Stability Violation. *DATE*, April 2009.

[9] J. Dorsey, S. Searles, M. Ciraula, S. Johnson, N. Bujanos, D. Wu, M. Braganza, S. Meyers, E. Fang, and R. Kumar. An Integrated Quad-Core Opteron Processor. *ISSCC*, pages 102–103, 2007.

[10] J.M. Rabaey, A.Chandrakasan, and B.Nikolic. *Digital Integrated Circuits, A design perspective, Second Edition, Chapter 3.* Pearson Education Asia Limited and Tsinghu University Press, 2004.

[11] K. Minami, M. Mizuno, H. Yamaguchi, T. Nakano, Y. Matsushima, Y. Sumi, T. Sato, H. Yamashida, and M. Yamashina. A 1 GHz portable digital delay-locked loop with infinite phase capture ranges. *ISSCC*, pages 350–351, 469, 2000.

[12] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanzawa, M. Ichida, and K. Nogami. Automated low-power technique exploiting multiple supply voltages applied to a media processor. *JSSC*, 33(3):463–472, Mar. 1998.

[13] P. Mahoney, E. Fetzer, B. Doyle, and S. Naffziger. Clock Distribution on a Dual-Core, Multi-Threaded Itanium-Family Processor. *ISSCC*, pages 292–293, 599, 2005.

[14] R. McGowen, C. A. Poirier, C. Bostak, J. Ignowski, M. Millican, W. H. Parks, and S. Naffziger. Power and temperature control on a 90-nm Itanium family processor. *JSSC*, 41(1):229–237, Jan. 2006.

[15] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. A Self-Tuning DVS Processor Using Delay-Error Detection and Correction. *JSSC*, 41(4):792–804, Apr. 2006.

[16] S. Ghosh. A Novel Delay Fault Testing Methodology using Low-Overhead Built-In Delay Sensor. *IEEE Trans. on CAD*, 25(12), 2007.

[17] S. Ghosh, S. Bhunia, and K. Roy. CRISTA: A New Paradigm for Low-Power, Variation-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation. *IEEE Trans. on CAD*, 26(11):1947–1956, Nov. 2007.

[18] S. Herbert and D. Marculescu. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. *ISLPED*, pages 38–43, 2007.

[19] Sun Microsystem Inc. OpenSPARC T1 Microarchitecture Specification. 2006.

[20] T. Xanthopoulos, D. W. Bailey, A. K. Gangwar, M. K. Gowan, A. K. Jain, and B. K. Prewitt. The design and analysis of the clock distribution network for a 1.2 GHz Alpha microprocessor. *ISSCC*, pages 402–403, 2001.

[21] V. Joshi, D. Blaauw, and D. Sylvester, . Soft-edge flip-flops for improved timing yield: design and optimization. *ICCAD*, pages 667–673, 2007.

[22] W. Zhao, and Y. Cao. New generation of Predictive Technology Model for sub-45nm early design exploration. *IEEE Trans. on Electron Devices*, 53(11):2816–2823, Nov. 2006. http://www.eas.asu.edu/∼ptm.

[23] Y. Jeon, J. Lee, H. Lee, K. Jin, K. Min, J. Chung, and H. J. Park. A 66-333-MHz 12-mW register-controlled DLL with a single delay line and adaptive-duty-cycle clock dividers for production DDR SDRAMs. *JSSC*, pages 2087–2092, 2004.