Adaptive Clocking Techniques for SoC Supply Droop Response in Predictive 7nm CMOS

Dan Fritchman, *Member, IEEE* and Wahid Rahman, *Member, IEEE*

***Abstract*—Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.**

***Index Terms*—Adaptive clocking, adaptive frequency, power efficiency, supply-droop mitigation, supply-voltage droop.**

I. Introduction

Power management techniques in modern system-on-chips (SoCs) are critical for energy-efficient processors ranging from data servers to mobile devices. SoC thermal dissipation constraints and energy-saving modes necessitate system-level power management to decrease the power supply or reduce the number of active processing cores. Such techniques exhibit a decrease (i.e. droop) in the SoC supply voltage (V_{DD}) due to: the controlled decrease of V_{DD} from a supply regulator or DC-DC converter; and the transient $L\frac{di}{dt}$ supply voltage ripple due to sudden current changes through package inductances when dynamically enabling or disabling on-chip processing cores. To maximize processing throughput, SoCs are designed to operate close to the maximum possible clock frequency (f_{MAX}) for the target (V_{DD}) with minimal guardbanding. Decreasing V_{DD} reduces the drain currents of CMOS transistors, thereby increasing propagation delays in the critical paths of digital logic. If sufficient timing margin is not available in the design, these critical paths can fail to meet timing and cause unrecoverable errors.

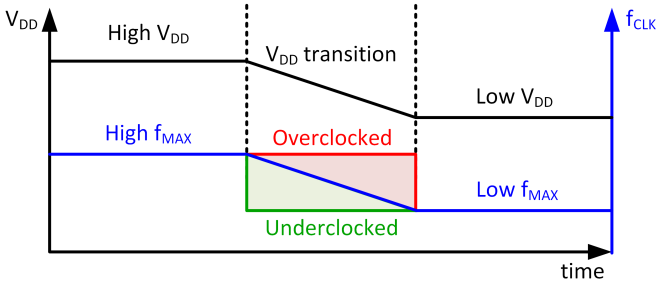

Fig. 1. Reduction of V_{DD} and corresponding reduction in f_{MAX} (adapted from [1]).

Date of publication March 21, 2020.

D. Fritchman and W. Rahman are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: dan_fritchman@berkeley.edu; wahid.rahman@berkeley.edu).

This work was supported by Professor Borivoje Nikolić.

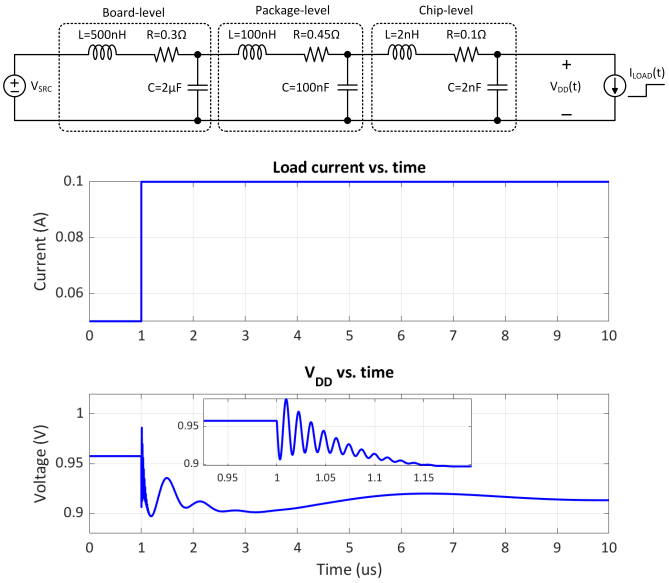

Fig. 2. Supply droop transients due to package inductances and load current (adapted from [2]).

To mitigate such failures, adaptive clock generation techniques have emerged in recent generations of high-performance SoCs [1]–[5]. Adaptive-clock systems detect transient supply events and dynamically adjust clock frequencies to not exceed a changing f_{MAX} as V_{DD} decreases. As Fig. 1 illustrates, a decrease from high to low V_{DD} to save power corresponds to a decrease in f_{MAX}, and the SoC clock frequency (f_{CLK}) must track this change without exceeding the correct f_{MAX} (overclocking) and with minimal guardbanding (underclocking) to minimize processing throughput loss during this change [1].

Transient response of adaptive clock techniques is also vital. The external package routes supplying power to the SoC impact supply voltage settling during changes to on-chip core utilization [2], as illustrated in Fig. 2. The range of inductances & capacitances in this network induces a medley of fast and slow time constants. Resilient adaptive clock generation circuits react quickly to fast ripples in V_{DD} ranging from 50-100MHz [2], [3] while also tracking slower transient settling at 1MHz and below [2], [3], [5].

Several schemes exist to realize supply-induced clock adaptation ranging from directly controlling the clock-synthesizing phase-locked loop (PLL) [1], [2] to modulating a tunable delay line or phase rotator downstream in an adaptive clock distribution (ACD) network [3]–[5]. In this work, we survey and contrast the adaptive PLL- and ACD-based mechanisms presented in [2] and [3], respectively. This brief is organized as follows: Section II provides an overview of adaptive clocking schemes and defines the scope of this work; Section III discusses the operating principles of mechanisms presented in [2] and [3]; Section IV provides a comparison of [2] and [3]; and finally Section V discusses planned work and concludes this brief.

II. Adaptive Clocking Schemes

Adaptive clocking systems include two fundamental components:

(a) a *power-supply sensor*, which measures and reports transient droops in supply voltage; and
(b) a *clock period actuator*, which modulates the system clock period in response to reports from the power-supply sensor.

During a supply drop, low-latency sensing and actuation is crucial to detect and adjust the clock frequency quickly to ensure error-free operation continues throughout the event. If and when V_{DD} recovers from this event to its nominal value, additional circuitry can assist with managing the corresponding f_{CLK} recovery.

ACD- and adaptive PLL-based systems differ in their implementation of the clock period actuator. ACD-based systems decrease the clock frequency by extending clock periods using an ACD circuit placed between the clock-synthesizing PLL and the global clock distribution network as illustrated in Fig. 3. A digital code $CTRL_{FCLK}$ from the power supply sensor controls the ACD circuit such that it extends the period of the synthesized clock, CLK_{PLL}, to the desired frequency given the sensed change in V_{DD}. This lower-frequency clock, CLK_{SOC}, is then distributed through the global clock network to the SoC processing cores. Examples of this ACD circuit include tunable-length delay [5] and phase rotators [3], of which the latter is discussed in Section III-B.

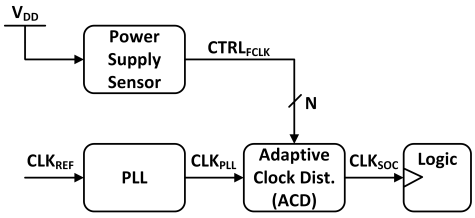

Fig. 3. Adaptive clock distribution (ACD) based system.

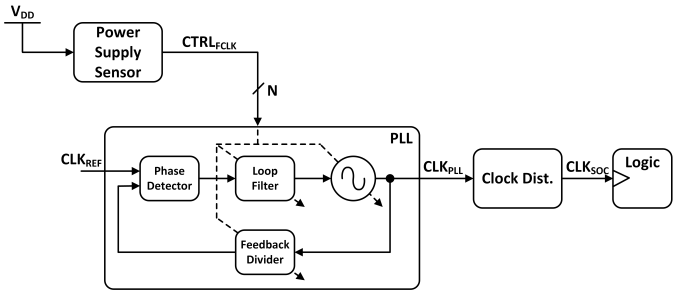

Fig. 4. Adaptive PLL-based system with possible intra-PLL control variants.

Adaptive PLL-based systems, in contrast, incorporate information from the power-supply sensor to dynamically change the behaviour of PLL sub-components. Typically, these schemes affect oscillator control on top of the traditional PLL loop. In [1] for example, the oscillator is controlled indirectly through a loop filter modified for droop response: when a droop is detected, the traditional proportional- and integral-loop filter switches to proportional-only to form a type-I PLL that tunes the oscillator to frequency lock without hazardous over- or underclocking. In [2], this idea is further extended by interrupting the PLL feedback loop to modify the oscillator directly, reacting to a supply droop event with relatively low latency. The feedback divider is then slowly modified to restore PLL lock. This latter work is further discussed in Section III-A.

The focus of this brief is to compare the design and effectiveness of clock period actuators in ACD and PLL-based schemes. While high-resolution power-supply sensors and their low-latency integration are critical to the overall adaptation time of such schemes, it remains beyond the scope of this work.

III. Operating principles

The two main works for comparison are the PLL- and ACD-based adaptive clocking schemes in [2] and [3] respectively. The operating principles of each are described in this section.

A. PLL-based adaptive clocking

The scheme reported in [2] presents a PLL-based adaptive clocking control for three SPARC processor cores in a 20nm CMOS testchip. A sense point placed near one of the SPARC cores transmits the core V_{DD} through low-impedance on-package (off-chip) routing to the on-chip power-supply sensor. This sensor converts V_{DD} droops into a thermometer-coded quantized digital signal, $q[n]$, by use two calibrated delay lines and a 8-bit time-to-digital converter (TDC).

A major contribution of [2] is to use the instantaneous value of q[n] to immediately reduce the PLL's oscillator frequency to correct for high-frequency droops and use a filtered version of $q[n]$ to correct for low frequency droops and ultimately re-lock the PLL feedback loop. As shown in Fig. 5 and Fig. 6, $q[n]$ is processed by the frequency control logic to generate two control signals: ΔF_{code} that can instantaneously respond to changes in $q[n]$ to directly change the digitally-controlled oscillator (DCO) frequency, and N_{code}^{sync} that relies only on a filtered value of $q[n]$ to track low-frequency droops and re-lock the PLL loop at the new frequency by changing the feedback division value. The minimum function and the non-linear filter together effectively construct an all-pass filter when V_{DD} decreases and a low-pass filter when V_{DD} increases. They allow the loop to actuate the clock with minimal latency during a V_{DD} droop event. During droop recovery, they low-pass filter V_{DD} transients and overshoots as the voltage ultimately settles to its nominal value.

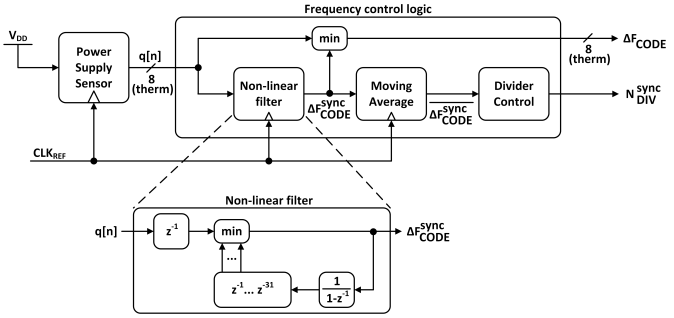

Fig. 5. Frequency control logic for PLL-based adaptive clocking in [2].

As fast response to high-frequency supply droops is an important criterion in [2], significant design efforts were made to ensure low-latency clock period actuation. The reported design target is 8× faster than the first-droop frequency of 50MHz. This permits a time window of 2.5ns from the time V_{DD} crosses the first supply sensor threshold to the time the SoC clock frequency is corrected. This narrow timing constraint is met by generating and propagating the critical oscillator control signal ΔF_{code} asynchronously. Crucially, the 8-bit thermometer-coded ΔF_{code} controls a DCO that can tolerate asynchronous changes in its code without glitching its output clock. The DCO reported in [2] is a bank of nine ring inverters with eight of the rings independently enabled by each thermometer bit of ΔF_{code}. Such a structure can tolerate asynchronous arrival of ΔF_{code} when

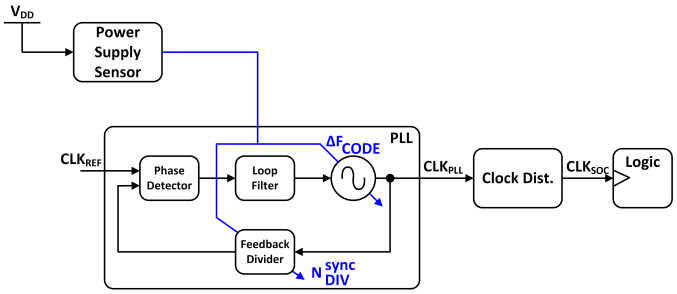

Fig. 6. Intra-PLL control for PLL-based adaptive clocking in [2].

suddenly decreasing frequency, as the case of a droop. Note that frequency increases are restricted: the combination of non-linear filter memory with the masking of minimum function ensures ΔF_{code} only increases one bit at a time after droop recovery [2]. This obviates synchronization of ΔF_{code}, reducing adaptation latency.

The reduction in adaptation latency directly correlates with reported experimental results. Fast response to droops allows for tighter guardbanding and therefore higher f_{MAX}. While [2] does not report the exact clock adaptation latency, the work improves f_{MAX} by 7.5%, from 4.65GHz without adaptive clocking to 5.00GHz with the proposed scheme. This is the largest frequency gain reported in compared works.

B. ACD-based adaptive clocking

The scheme reported in [3] presents a ACD-based adaptive clocking control for AMD's Steamroller CPU in 28nm CMOS. The power supply sensor consists of a DLL-based droop detector acting as a delay line connected to the SoC core V_{DD}: as V_{DD} changes, the DLL phases change with respect to a regulated PLL's output clock. A programmable threshold selects one of four DLL phases to compare against the reference PLL phase to signal a 1-bit droop activity.

Fig. 7 illustrates the reported scheme. A DLL-based phase rotator realizes the ACD circuit. When a droop activity is detected and $DroopDetected$ is asserted, phase rotation between the 40 DLL phases occurs to effectively "stretch" the clock period and reduce the clock frequency. To ensure glitchless phase rotation, small positive pre-programmed phase increments occur at each step, and the phase rotator rotates through all 40 phases before continuing the cycle. Rotation can be bypassed entirely by de-asserting $DroopEnable$.

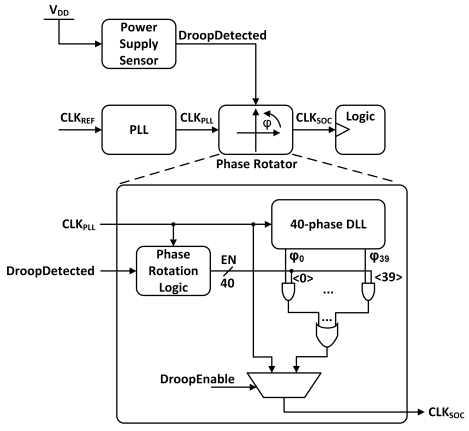

Fig. 7. Frequency control logic for ACD-based adaptive clocking in [3].

This scheme reports to target fast initial droops of 100MHz and slower 1-5MHz secondary droop transients by means of the programmed droop detector threshold [3]. As the detector signal is binary, the ACD-based reduces the SoC clock frequency by a fixed amount (phase rotator steps through a fixed phase step per cycle) or does not change the frequency at all. Voltage recovery is triggered when V_{DD} crosses back beyond the detector threshold. The work reports a total adaptation latency of 3 cycles at 3.4GHz from droop detection to clock frequency reduction. While experimental results do not report frequency gain as in [2], silicon results demonstrated a reduction of 3-6% in V_{MIN}, the minimum supply voltage tolerable at a given frequency for the SoC. No comparison was provided with prior works.

IV. Comparison of ACD and PLL Actuators

TODO (Dan's draft + summarize comparison between two from above.) Comparison table? Metrics (frequency gain, VMIN reduction, detector type, etc...)

V. Conclusion

TODO

References

[1] F. Ahmad *et al.*, "A 0.5-9.5-GHz, 1.2-μs Lock-Time Fractional-N DPLL with $\pm$1.25%UI Period Jitter in 16-nm CMOS for Dynamic Frequency and Core-Count Scaling," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 21–32, Jan. 2017.

[2] T. Hashimoto *et al.*, "An Adaptive-Clocking-Control Circuit with 7.5% Frequency Gain for SPARC Processors," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 1028–1037, Aug. 2018.

[3] K. Wilcox *et al.*, "Steamroller Module and Adaptive Clocking System in 28 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 24–34, Jan. 2015.

[4] M. S. Floyd *et al.*, "Adaptive clocking in the POWER9^{TM} processor for voltage droop protection," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, USA, Feb. 2017, pp. 444–445.

[5] K. A. Bowman *et al.*, "A 16nm All-Digital Auto-Calibrating Adaptive Clock Distribution for Supply Voltage Droop Tolerance Across a Wide Operating Range," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 8–17, Jan. 2016.