

# Modeling and Analysis of Adaptive Frequency Synthesis for Supply Droop Mitigation

Sunjin Choi, *Student Member*  
sunjin\_choi@berkeley.edu

**Abstract**—Dynamically changing clock frequencies to guard system against timing failure is a promising technique for power/performance optimization in modern commercial processors. Also known as Adaptive Clocking scheme, it can reduce Vdd guardband and thus the minimum achievable supply voltage  $V_{min}$  while guaranteeing error-free operations. Adaptive Frequency Synthesis is one of such techniques to mitigate the effect of high-frequency supply noise at hundreds of megahertz by directly adapting the PLL-driven clock at the instant of droop detection. However, previous studies lack system-level analysis of such systems except that presented in [1]. In this paper, we present a system model for AFS systems by extending the model in [1] and accommodating a simplified PLL model in Verilog-A. Based on the model, we derive the relationship between design parameters like PLL loop bandwidth, frequency step, clock insertion delay and system performance metric  $V_{min}$ .

**Index Terms**—Adaptive Frequency Synthesis (AFS), Adaptive Clocking (AC), Power Supply Noise (PSN)

## I. INTRODUCTION

Supply voltage droop imposes a critical limitation to processor power optimization while ensuring the timing constraints to be met across wide operating conditions. While supply droop transients occur infrequently during the system operation, designers should allocate a certain amount of supply margin to tolerate any worst-case supply variations. However, it is also beneficial to reduce the supply margin as reducing Vdd directly translates into power efficiency gain. Alternative approaches to reduce excess supply margin is to adaptively tune the clocking circuits during a droop event to avoid potential timing failures. One of the popular implementations is Adaptive Frequency Synthesis (AFS) which directly modulates PLL-driven clocks to compensate for the delay changes at critical paths. In this paper, we propose a model of AFS system to study the effects of various design parameters on the system performance.

A challenge with designing adaptive clocking circuits is to ensure correct functionalities under a large supply droop of a few millivolt at a few hundreds of megahertz speed. Traditionally, Vdd guardband was employed to guard against supply variations without intricate clocking schemes. Introduction of adaptive techniques reduced Vdd guardband to enhance the power efficiency of the processors, as illustrated in Fig. 1. However, the requirements of adaptive clocking are getting more stringent with the processor generation. For example, it should be (1) fast enough to mitigate nanosecond-scale fast droops, (2) implemented with low area/power overhead, (3) calibrated across DVFS modes.

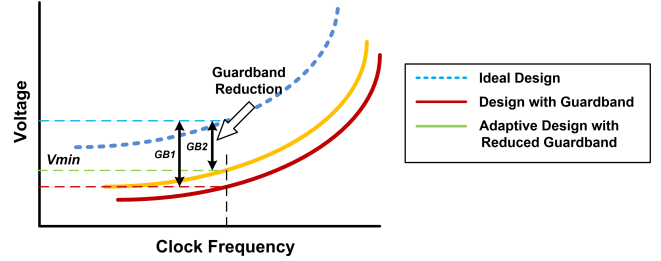


Fig. 1: Guardband Reduction by Adaptive Design

Numerous techniques have been proposed over 15 years to meet those requirements like Clock Stretcher (CS) [2]–[4], Adaptive Clock Distribution (ACD) [5], [6], Proactive Clock Gating (PCG) [7], [8], etc. Among these techniques, AFS can minimize the area/power overhead as it can be implemented with a droop detector and corresponding PLL controls.

As adaptive clocking system in general is required to quickly adapt clock frequency to a supply droop and slow-down of datapath delay, it is important to model the response latency of the systems and correlate various design parameters to the delay. Time Domain Model proposed in [1] identifies the design constraints of the subsystems by modeling the response latency as a sum of detection and response delays. Their impact on the system is then investigated in terms of system performance metric  $V_{min}$ , or minimum achievable supply voltage. Also a phenomena called Clock-Data Compensation (CDC) is studied in [9], [10] with a small signal delay model to analyze the effect of various design parameters and served as a basis of ACD circuit designs.

In this paper, we extend the model in [1] to AFS systems and study the relationship between system performance metrics ( $V_{min}/F_{max}$ ) and system design parameters such as response delay and PLL parameters. While our proposed model is a straightforward extension of that presented in [1], it allows the exploration of the system-level impact of PLL design parameters such as loop bandwidth and size of the frequency step. Also, we will re-examine the effect of design insights described in [1] with a simple supply droop model to provide analytical flavors to obtained results. Moreover, we aim to highlight the key design considerations regarding bandwidth constraints of each sub-component that can serve as a guideline for AFS system designs.

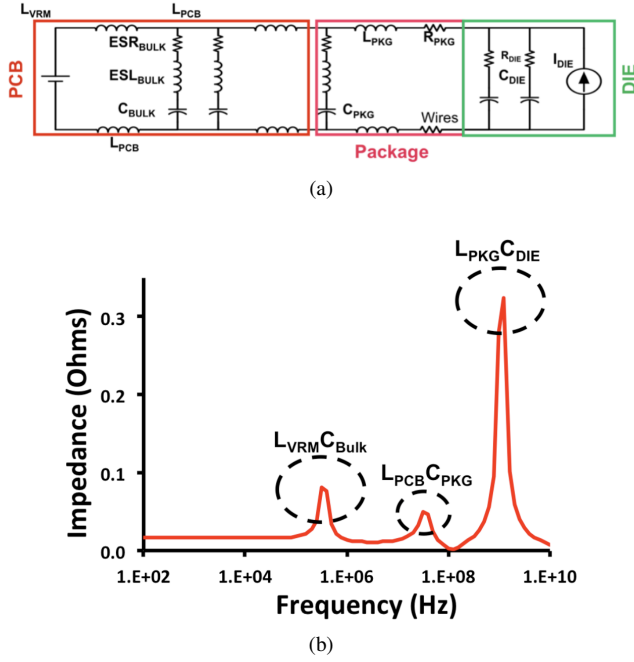


Fig. 2: (a) Simplified PDN Model, (b) PDN Impedance Profile with Resonances Highlighted [11].

## II. BACKGROUND

Popular approaches to mitigate the effect of supply noise can be classified as: Power Delivery Network (PDN) design, adaptive clocking circuits and architectural-level error recovery schemes. The most straightforward one is to place enough capacitors at each board/package/die-level to make supply impedance as low as possible across a wide range of frequencies [12]. However, it is generally impractical to use enough bulky capacitors to completely flatten out supply impedance. Another approach to address the supply noise issue, called Resilient Design is to detect and correct the errors at the architectural state of the processor through error recovery processes such as instruction replay [13]. Though a resilient design approach can resolve the inherent bandwidth limitations of adaptive clocking circuits, it is challenging to cover all the possible scenarios of architectural state corruption. Here we focus on the Adaptive Clocking designs where supply droop mitigation is implemented only inside the clock generation/distribution circuits.

### A. Supply Droop

Dominant factor of supply droop is resonance peaks at the power supply impedance, where board/package/die-level inductances and capacitances induce resonance peaks at multiple frequencies and amplify injected noise current from surrounding digital systems as seen from Fig. 2a. Fig. 2b shows the resonance peaks from typical power supply impedance profiles that occur at three distinct frequencies. For example, first-order resonance happens at approximately 100MHz, while second- and third-order resonances at  $\sim 1$ MHz

and  $\sim 10$ kHz, each associated with  $L_{pkg}C_{die}$ ,  $L_{pcb}C_{pkg}$ ,  $L_{reg}C_{bulk}$  respectively [11]. As  $\Delta V = \Delta I * Z$ , both current noise injection and supply impedance profile determine the magnitude and frequency of supply voltage droop. Hence supply voltage droop is actually a function of workload variations which makes only post-silicon measurements valid for the droop characterizations [12]. Nevertheless, supply droop is often simplified as a combination of the first, second and third droop each corresponding to resonant peak.

Mitigating supply droop is challenging in both magnitude and frequency perspective. In frequency perspective, the first droop in 100MHz has also the largest magnitude of three and forces the circuits to detect and respond to it in a few nanoseconds. On the other hand, in magnitude perspective, technology scaling drives  $di/dt$  to a larger value and hence the larger droop, since  $di$  is proportional to the transistor counts and  $1/dt$  is proportional to the clock frequencies [14]. Moreover, as described in [15], core-to-core interactions can cause perfect storms of supply droop, which is approximately 1.2x larger in magnitude and 1.9x steeper in slope.

### B. State-of-the-art Adaptive Clocking Circuits

AMD's Zen Processor adopted the CS scheme to change the clock frequency by continuously switching clock phases and stretch the clock at a droop event [2]–[4]. However, the main drawback of this scheme is synchronization overhead to the response latency, where a synchronizer block should be added in between the droop detector and the phase picker logic to safely use the output signal from the droop detector. Also, additional phase picking logic adds jitter to the output clock by 0.5~1% UI. In Zen Processor, the scheme is revised to have coarse- and fine-grain clock-stretchers to reduce the overall response time.

Another interesting technique proposed by Bowman [5], [6] is ACD, where CDC is exploited to prevent the timing failure for multiple cycles after the onset of supply droop. CDC is a phenomena that the clock edge is pushed out by slow-down of clock distribution path proportionally to slow-down of the datapath delay both at a Vdd droop. With CDC, we can secure a sufficient amount of time to respond to the timing margin degradation which is delayed by several cycles. This scheme can also be auto-calibrated [6], removing tester calibration overheads. Moreover, It is fast as it does not require a synchronizer block between detection and adaptation blocks. However, clock buffers and clock gating cells should be fine-grain controlled which is expensive in terms of clock distribution designs.

Further reduction of response time is possible by implementing the system with proactive response rather than reactive. Recent IBM's z15 chip [8] combined proactive voltage droop detection with traditional Critical Path Monitor to further reduce the detection response time. However, as pointed out in [7], proactive detection is prone to mispredictions due to an unpredictable dataflow execution in general-purpose CPUs. Thus the Qualcomm's PCG system in its recent Hexagon DSP is implemented based on the



is modeled as an FO-1 inverter chain with 40fF side loads distributed along the chain with approximately 10% slack at the nominal supply voltage 0.7V.

#### A. Time Delays in Adaptive Frequency Synthesis

As explained above, delays in clock frequency adaptation is one of the key parameters that can determine the droop mitigation capabilities of the system. Three major components of delays are: 1) droop detector delay, 2) clock insertion delay from central-PLL, 3) clock insertion delay from leaf-PLL to critical path. As the major functionalities of droop detection is to determine if the droop event has actually happened and when the event has happened, we focus on the latter one and abstract the droop detector as a simple time delay block. Therefore, droop detection and clock tree insertion is simplified into a delay block which can be represented as `absdelay` in Verilog-A. Typical numbers for droop detector delay spans 2-3 cycles and for clock tree insertion delay spans 5-10 cycles, we set the nominal values for those delays as 0.5ns, 1.3ns at 5GHz clock frequency.

#### B. First-order Supply Droop Model

Abstracting supply droop by its first-order slew response is useful in that it can quickly reveal if the system can adapt its clock frequency with respect to the fastest slope in droop event. First-order slew response, at the speed of approximately 100MHz, has usually been the major hurdle in various adaptive clocking schemes. Previous implementations on adaptive frequency systems claim to have superb droop mitigation capabilities at such high frequencies, first-order abstraction of supply droop may help reveal those benefits in a simpler fashion. A more realistic evaluation would include droop response of all three orders, or supply profile from actual processor and workload model to calculate the performance gain at the actual system deployment.

Major parameters of the first-order slewing model is the magnitude and slew rate of the droop. Magnitude of the droop response determines the minimum operating supply voltage  $V_{min}$  by  $V_{min} = V_{dd} - V_{droop,max}$ . Slew rate can be expressed as  $SR = |dV/dt| = 2\pi F_{res} V_{droop,max}$  where  $F_{res}$  is the first-order resonance frequency of the given PDN impedance. As  $|dV/dt| = V_{droop,max}/t_{droop}$  with  $t_{droop}$  being time window for the slew response, we can arrange the equation for  $t_{droop}$  into  $t_{droop} = 1/(2\pi F_{res})$ . As  $F_{res}$  typically lies between 60-100MHz, we can assume the nominal value for  $t_{droop}$  as 1.5ns for our testbenches.

#### C. PLL Frequency Adaptation Model

As from IBM and Intel's implementations, PLL modulations at AFS systems can be abstracted into initial output clock frequency steps and PLL gradually smoothing out the step response. For example, IBM implemented the modulation path at DCO input, shifting the DCO code as the response of droop detection outputs. Intel, on the other hand, directly added DCO devices and stepped down the clock frequency and further reduced the DCO response delay. Both implementations share

a common feature that direct modulation is done in a way that DCO outputs are first modulated and then PLL dynamics reacts to the modulation.

Thus we can model PLL frequency adaptation with simple s-domain representations from frequency step input to PLL frequency response, as commonly done in PLL design practices. Let target frequency step be  $F_{step}$ , and let PLL damping factor and natural frequency as  $\zeta$  and  $w_n$ , we can write the s-domain PLL transfer function as follows:

$$\frac{F_{out}}{F_{in}} = \frac{s^2 F_{step}}{s^2 + 2\zeta w_n s + w_n^2} \quad (1)$$

Nominal values of  $\zeta$  and  $w_n$  would be set by PLL design practices according to the jitter, lock time/range specs. Our testbenches assumes  $\zeta$  of 1.2 and  $w_n$  of 2MHz as a typical design target. We experimented over two  $F_{step}$  values, 156MHz and 78MHz, or 3.125% and 1.5625% of 5GHz clock frequency where the former one is derived from [19].

Modeling PLL response in Verilog-A allows for the direct integration into the SPICE-based testbenches, but care has to be taken to implement necessary Verilog-A blocks. Our PLL model implementation consists of three parts: s-domain transfer function from step input to output clock frequency, frequency-to-phase block and phase-to-voltage block. The first two blocks can simply be implemented with Verilog-A primitive functions. Therefore, s-domain transfer function is implemented in `laplace_nd` function where frequency-to-phase block is implemented in `idtmod` function. However, it is not trivial to implement phase-to-voltage block since we should generate clock pulses aligned with the input phase values. Fig. 6 shows the implemented phase-to-voltage block [20], which first encodes the phase information in generated sinusoid waves and then carves the sinusoids into the pulse-like shapes with `tanh` function.

### IV. SIMULATION RESULTS

Fig. 7 demonstrates the block diagram for the main testbench and  $V_{min}$  simulation methodology. In this section, as highlighted in the block diagram, we present the simulation results that demonstrates the effect of three key design parameters on system performance: 1) size of the frequency step  $F_{step}$ , 2) adaptation delay  $T_{DD} + T_{tree}$ , 3) PLL parameters  $\zeta$  and  $w_n$ . Their impact on the system will be evaluated in terms of  $V_{min}$ , which is the performance metric that

```

module phase2v (phin, vout);
    input phin;
    output vout;

    // parameter definitions omitted for brevity
    analog begin
        v(sine) <+ cos(2*M_PI*freq*$abstime + V(phin));
        v(vout) <+ voffset +
                    vamplitude*tanh(alpha*V(sine));
    end
endmodule

```

Fig. 6: Phase-to-Voltage Block written in Verilog-A



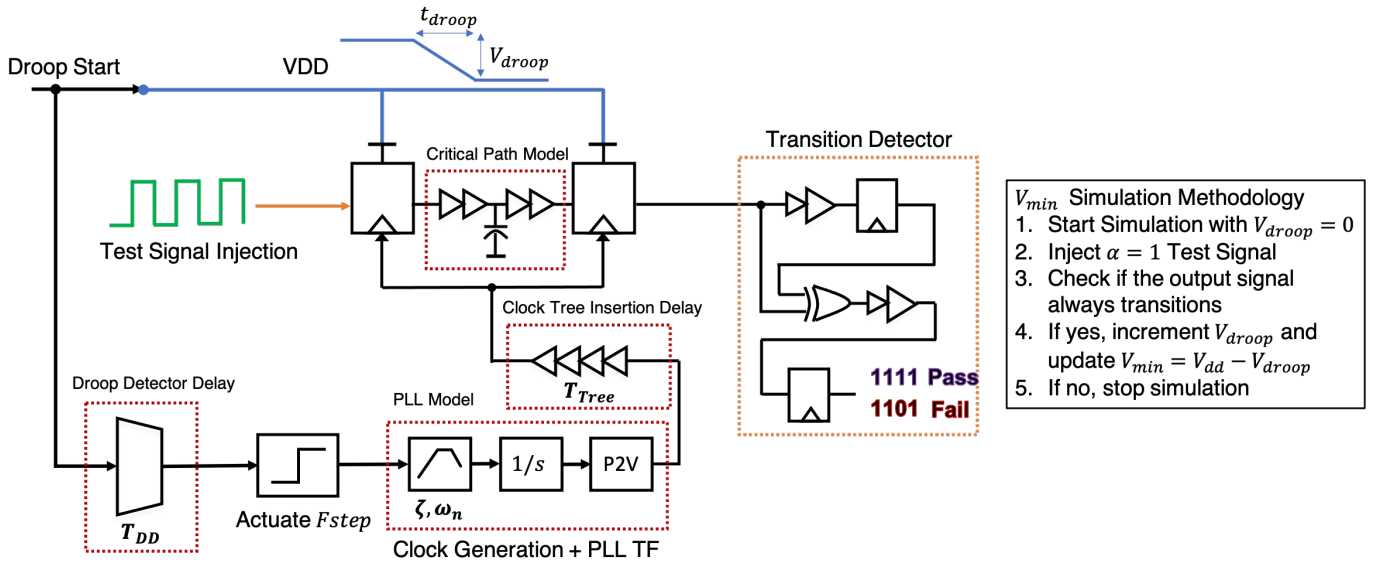


Fig. 7: Block Diagram for Testbench Setup and  $V_{min}$  Simulation Methodology

is directly related to processor power efficiency and that adaptive clocking schemes eventually aim to optimize.  $V_{min}$  is simulated by injecting a test signal with activity factor equals 1 and see if the output of the end-point FF always makes a transition. By sweeping over the droop magnitude and observing the transition detector output, we can simply identify the minimum operating supply voltage  $V_{min}$ . Unless stated otherwise, we use the default parameters of  $F_{step}$  3.125%,  $T_{dd}$  0.5ns,  $T_{tree}$  1.3ns,  $\zeta$  1.2,  $w_n$  2MHz,  $t_{droop}$  1.5ns as explained at the previous section.

#### A. Model Validation

In this subsection, we check the model validity by comparing the result against the baseline architecture, where clocks are distributed directly from central-PLL. Fig. 8 is the timing diagram that compares the resulting slacks from two different clocking schemes. In this example, we applied a droop magnitude of 80mV and overall adaptation delay of 0.6ns to better illustrate the AFS operation. The baseline architecture with central-PLL generated clocks which is not compatible with the frequency adaptation since the droop itself varies from chip-to-chip and core-to-core. In the same context, frequency adaptation should be employed with per-core PLL clocking schemes. By comparing two schemes in terms of slack improvement, We can see from the Fig. 8 that simulation result aligns with our intuition that adaptive frequency can improve slack degradation at the supply droop event. Baseline architecture results in the minimum slack of 1.49ps at the measuring window, while frequency adaptation architecture improves it to 27.5ps slack.

Also, one interesting observation from Fig. 8 is that slack improves right after the PLL response. At 0.6ns after the droop event starts, which is the time PLL responds to the frequency step, slack improves from baseline to adaptation architecture

by 10  $\tilde{1}$ 5ps. This aligns with our s-domain analysis from Eq. 1. Calculating the step response of Eq. 1, we can see that PLL output clock frequency changes instantaneously with the step input and followed by small ringings, resulting in immediate slack improvement with PLL receiving the step.

#### B. Effect of Frequency Step Size

In this subsection, we compare the effect of two different frequency step sizes, 3.125% and 1.5625% of 5GHz clock frequency on the system performance. Table I demonstrates such comparison in terms of minimum slack at 78mV supply droop and overall  $V_{min}$  improvement. It is simulated with a total adaptation delay of 0.6ns to better illustrate the difference between two. According to the table, having a 3.125% frequency step improves minimum slack by 3.18ps and  $V_{min}$  improvement of 5mV as compared to the case of 1.5625% frequency step. While the improvement is not significant in our simulation results, it clearly shows that  $V_{min}$  scales with the frequency step size, thus implying two design considerations. First, using a large frequency step can help the system adapt to the larger voltage droop. Hence two-step frequency adaptation proposed in [19] can be useful in this context which uses 3.125% step for a smaller droop and 6.25% step for a larger droop. Second, as  $V_{min}$  is sensitive to the frequency step size, we should also employ a proper calibration method to stabilize frequency step sizes under DVFS operating conditions.

TABLE I: Effect of Frequency Step on System Performance

Frequency Step (%)	Minimum Slack (ps)	$V_{min}$ (mV)
3.125	8.65	5
1.5625	5.57	10

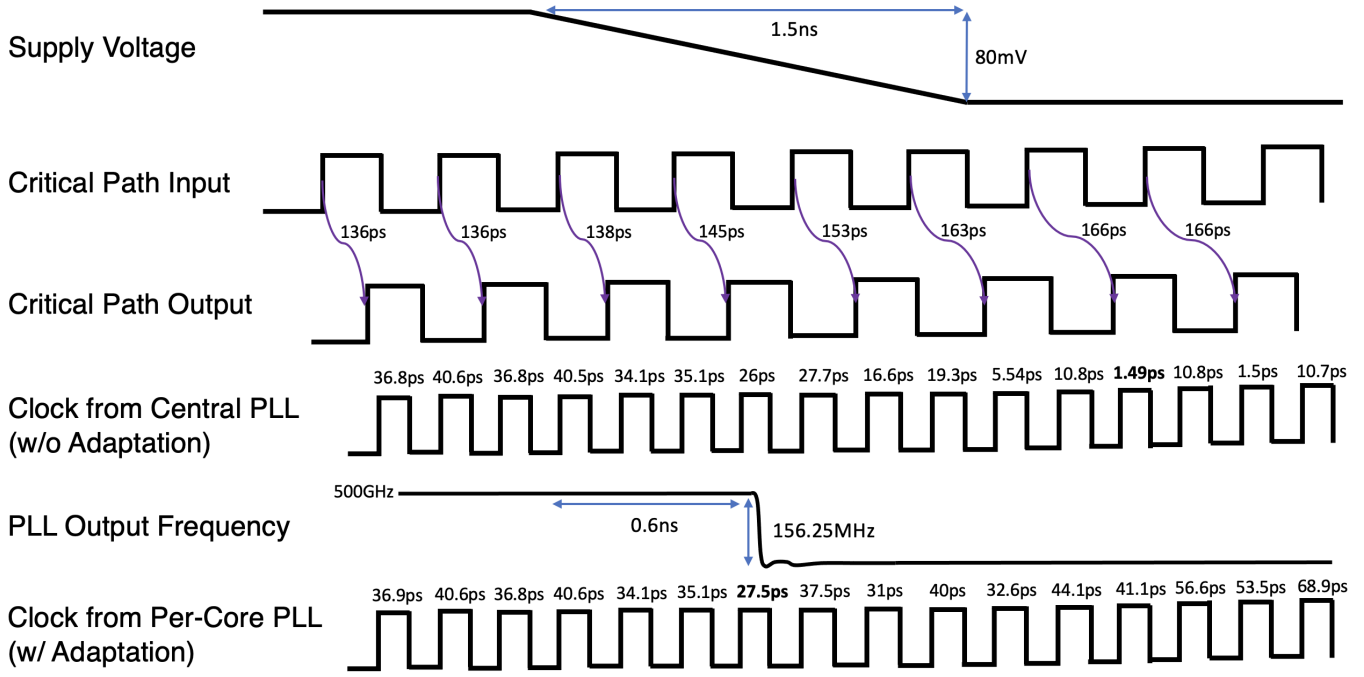


Fig. 8: Timing Diagram Comparing PLL Frequency Adaptation with Baseline Central-PLL Clocking

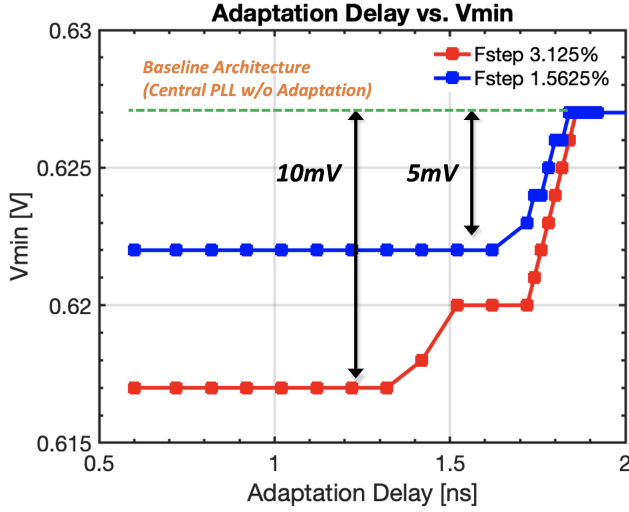


Fig. 9: Effect of Adaptation Delay on  $V_{min}$

### C. Effect of Adaptation Delay

Fig. 9 demonstrates the relationship between the performance metric  $V_{min}$  and adaptation delays, which is a sum of droop detector response time and clock tree insertion delay. With 3.125% frequency step from 5GHz clock,  $V_{min}$  almost linearly degrades from adaptation delay of 1.3ns to 1.8ns.  $V_{min}$  degradation then saturates with adaptation delay larger than 1.8ns where frequency adaptation becomes ineffective as compared to the baseline case. Moreover,  $V_{min}$  does not improve beyond 1.3ns adaptation

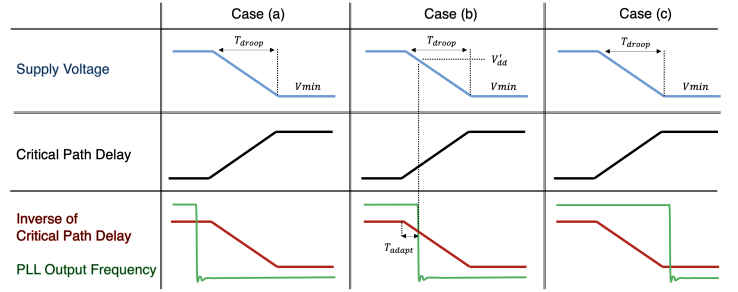


Fig. 10: Illustration for Delay-to- $V_{min}$  Analysis

delay. The case of 1.5625% frequency step also shows the similar behavior at 1.6ns and 1.8ns delays. In both cases, we can see that the delay-to- $V_{min}$  relation can be divided into three regions: upper and lower side of saturation regions and a linear region in between. This result suggests that we should first look for lower and upper saturation points for the given system and design droop detector and clock tree delays to be smaller than the lower saturation point.

This result can be understood intuitively with the aid of Fig. 10. First, we assume delay sensitivity to supply voltage is -1, and PLL output frequency responds instantaneously to the step stimuli at AFS schemes as drawn in Fig. 10. Let  $V_{dd,min}$  corresponds to the minimum supply voltage that critical path timings are met at 5GHz clock frequency without any adaptations. Also let  $V'_{dd,min}$  correspond to the minimum supply voltage at  $(5\text{GHz} - F_{step})$  clock frequency without adaptations. We first analyze the lower side of

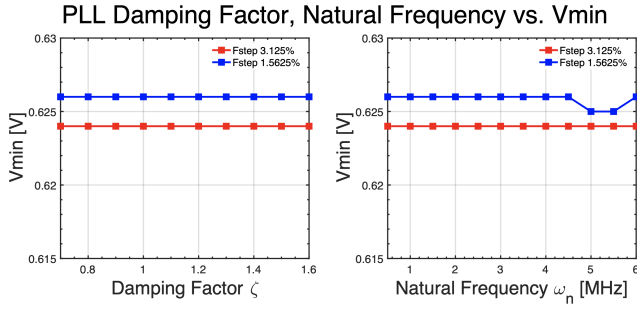


Fig. 11: Effect of PLL parameters  $\zeta$ ,  $\omega_n$  on  $V_{min}$

saturation where  $V_{min}$  does not improve with decreasing adaptation delays. It corresponds to the case (a) of Fig. 10. Since PLL output frequency is already set to  $(5\text{GHz} - F_{step})$  well before supply voltage reaches its minimum,  $V_{min}$  equals  $V'_{dd,min}$  independent of adaptation delays. Next, we analyze the upper side of saturation where  $V_{min}$  does not degrade with increasing adaptation delays which corresponds to the case (c) of Fig. 10. Since PLL steps down its output frequency after supply voltage has reached its minimum, but PLL frequency is always smaller than the nominal clock frequency 5GHz. Therefore  $V_{min}$  equals to  $V_{dd,min}$  which is also independent of adaptation delays. Finally, linear delay-to- $V_{min}$  relationship can be explained with a simple math. Let PLL adaptation happens  $T_{adapt}$  after the droop event. Since the output clock frequency is 5GHz before PLL adaptation, to guarantee error-free operations before  $T_{adapt}$ ,  $V'_{dd}$  in case (b) should be at least  $V'_{dd,min}$ . Therefore, a simple ratio  $T_{adapt} : T_{droop} = (V_{dd} - V'_{dd}) : (V_{dd} - V_{min})$  from Fig. 10 verifies the linear relationship between the adaptation delay and  $V_{min}$  as shown in Fig. 9.

#### D. Effect of PLL Design Parameters

Fig. 11 demonstrates the effect of PLL parameters on  $V_{min}$ . It shows that there is a negligible impact by changes in damping factor and natural frequency. This can also be understood with Fig. 10, that  $\zeta$  and  $\omega_n$  contribute only to the PLL response after the initial frequency step-down and do not affect the critical path slack. Therefore it suggests that PLL design itself can be optimized in terms of jitter and lock time/range requirements without affecting  $V_{min}$  performance. However in reality,  $\zeta$  and  $\omega_n$  should be carefully designed to make sure that it does not impact the frequency adaptation performance under DVFS operating conditions.

#### V. CONCLUSION

In this paper, we proposed a SPICE-based system-level simulation model for Adaptive Frequency Synthesis systems. Extension of ARM's time domain model to AFS systems with Verilog-A based PLL models and simplifying assumptions allowed the evaluation of system performance metric  $V_{min}$  in terms of system design parameters. Using the system model, we investigated the effect of design parameters on

$V_{min}$  that led us to the design insights on adaptation delays, PLL dynamics and frequency step sizes. We expect that the simulation model can be used for the design space explorations of AFS systems, revealing the design tradeoffs between other adaptive clocking techniques like CDC and Clock-Stretcher. Future studies would include more realistic frequency steps and performance variations under PVT and DVFS to provide more detailed design guidelines.

#### REFERENCES

- [1] P. N. Whatmough, S. Das, and D. M. Bull, "Analysis of adaptive clocking technique for resonant supply voltage noise mitigation," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2015, pp. 128–133.
- [2] A. Grenat, S. Pant, R. Rachala, and S. Naffziger, "5.6 adaptive clocking system for improved power efficiency in a 28nm x86-64 microprocessor," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 106–107.
- [3] K. Wilcox, R. Cole, H. R. Fair III, K. Gillespie, A. Grenat, C. Henrion, R. Jotwani, S. Kosonocky, B. Munger, S. Naffziger *et al.*, "Steamroller module and adaptive clocking system in 28 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 24–34, 2014.
- [4] T. Singh, A. Schaefer, S. Rangarajan, D. John, C. Henrion, R. Schreiber, M. Rodriguez, S. Kosonocky, S. Naffziger, and A. Novak, "Zen: An energy-efficient high-performance  $\times 86$  core," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 102–114, 2017.
- [5] K. A. Bowman, C. Tokunaga, T. Karnik, V. K. De, and J. W. Tschanz, "A 22 nm all-digital dynamically adaptive clock distribution for supply voltage droop tolerance," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 4, pp. 907–916, 2013.
- [6] K. A. Bowman, S. Raina, J. T. Bridges, D. J. Yingling, H. H. Nguyen, B. R. Appel, Y. N. Kolla, J. Jeong, F. I. Atallah, and D. W. Hansquine, "A 16 nm all-digital auto-calibrating adaptive clock distribution for supply voltage droop tolerance across a wide operating range," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 8–17, 2015.
- [7] V. K. Kalyanam, E. Mahurin, K. A. Bowman, and J. A. Abraham, "A proactive system for voltage-droop mitigation in a 7-nm hexagon<sup>TM</sup> processor," *IEEE Journal of Solid-State Circuits*, 2020.
- [8] D. Wolpert, C. Berry, B. Bell, A. Jatkowski, J. Surprise, J. Isakson, O. Geva, B. Deskin, M. Cichanowski, D. Hamid *et al.*, "Cores, cache, content, and characterization: Ibm's second generation 14-nm product, z15," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 98–111, 2020.
- [9] K. L. Wong, T. Rahal-Arabi, M. Ma, and G. Taylor, "Enhancing microprocessor immunity to power supply noise with clock-data compensation," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 749–758, 2006.
- [10] D. Jiao, J. Gu, and C. H. Kim, "Circuit design and modeling techniques for enhancing the clock-data compensation effect under resonant supply noise," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 10, pp. 2130–2141, 2010.
- [11] S. Das, P. Whatmough, and D. Bull, "Modeling and characterization of the system-level power delivery network for a dual-core arm cortex-a57 cluster in 28nm cmos," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2015, pp. 146–151.
- [12] R. Bertran, A. Buyuktosunoglu, P. Bose, T. J. Slegel, G. Salem, S. Carey, R. F. Rizzolo, and T. Strach, "Voltage noise in multi-core processors: Empirical characterization and optimization opportunities," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2014, pp. 368–380.
- [13] K. A. Bowman, "Adaptive and resilient circuits: A tutorial on improving processor performance, energy efficiency, and yield via dynamic variation," *IEEE Solid-State Circuits Magazine*, vol. 10, no. 3, pp. 16–25, 2018.
- [14] N. Kurd, P. Mosalikanti, M. Neidengard, J. Douglas, and R. Kumar, "Next generation intel core<sup>TM</sup> micro-architecture (nehalem) clocking," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1121–1129, 2009.

- [15] I. Pierce, J. Chuang, C. Vezirtzis, D. Pathak, R. Rizzolo, T. Webel, T. Strach, O. Torreiter, P. Lobo, A. Buyuktosunoglu *et al.*, “26.2 power supply noise in a 22nm z13™ microprocessor,” in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 438–439.
- [16] M. S. Floyd, P. J. Restle, M. A. Sperling, P. Owczarczyk, E. J. Fluhr, J. Friedrich, P. Muench, T. Diemoz, P. Chuang, and C. Vezirtzis, “26.5 adaptive clocking in the power9™ processor for voltage droop protection,” in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 444–445.
- [17] P. Mosalikanti, Q. Wang, K.-Y. J. Shen, M. Neidengard, S. F. S. Farooq, V. Grossnickle, and N. Kurd, “29.3 80ns fast-lock 0.4-to-6.5 ghz clock generator with self-referenced asynchronous adaptive droop mitigation,” in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 408–410.
- [18] C. R. Lefurgy, A. J. Drake, M. S. Floyd, M. S. Allen-Ware, B. Brock, J. A. Tierno, and J. B. Carter, “Active management of timing guardband to save energy in power7,” in *2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2011, pp. 1–11.
- [19] C. Gonzalez, M. Floyd, E. Fluhr, P. Restle, D. Dreps, M. Sperling, R. Rao, D. Hogenmiller, C. Vezirtis, P. Chuang *et al.*, “The 24-core power9 processor with adaptive clocking, 25-gb/s accelerator links, and 16-gb/s pcie gen4,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 91–101, 2017.
- [20] J. Kim, K. D. Jones, and M. A. Horowitz, “Variable domain transformation for linear pac analysis of mixed-signal systems,” in *2007 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2007, pp. 887–894.