Quasi-Resonant Clocking: A Run-time Control Approach for True Voltage-Frequency-Scalability

Visvesh Sathe
Department of Electrical Engineering
University of Washington, Seattle
sathe@uw.edu

ABSTRACT

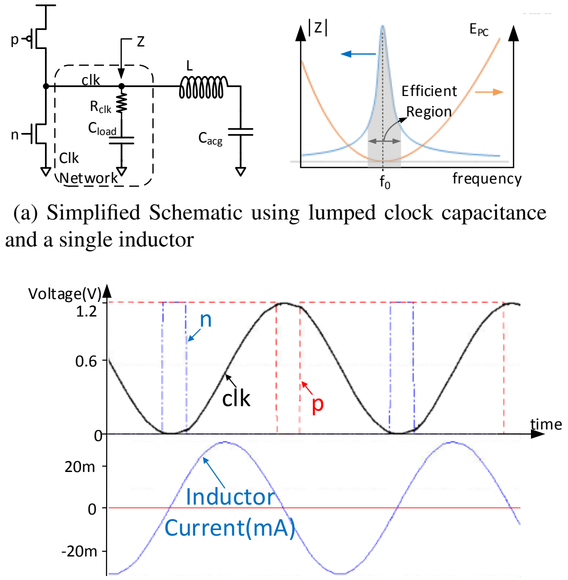
Resonant clocking has emerged as a promising approach for achieving energy-efficiency in high-performance digital systems. However, the limited frequency range of efficient resonant clocking operation restricts its applicability in widely-used Dynamic Voltage and Frequency Scaling (DVFS) systems. Existing frequency-scalable resonant clocking implementations are either not voltage-scalable, or provide only modest frequency range extension. This paper presents a *true* voltage and frequency-scalable quasi-resonant clock architecture. Simulations on a 64-bit pipelined multiply-accumulate unit in 65nm CMOS demonstrate continuous frequency scalability over 2–200MHz. Efficient operation during dynamic voltage frequency-scaling is demonstrated over 0.8V–1.3V, resulting in a 54% energy-per cycle reduction over conventional distributions.

1. INTRODUCTION

Clock energy dissipation continues to play a significant role in determining the energy-efficiency of a wide range of digital systems, from high performance microprocessors [8–10, 13] to ultra-low power digital circuits employing aggressive pipelining [7]. Following several prototype demonstrations [1–3, 6, 14–16, 18], resonant clocking has emerged as a promising technique to reduce the substantial power dissipated in commercial processor clock distributions [5, 10, 13].

A reliance on mesh-like structures to provide a common point in global clock distributions for low skew and high race-immunity is pervasive among high-volume, high-performance microprocessors [9–11, 17]. With its significant load capacitance, this clock mesh structure accounts for a sizable fraction of average power dissipation of the processor [12, 17].

The main concept behind resonant clocking is to employ inductance to achieve efficient LC resonance, enabling efficient oscillation of the capacitive global clock distribution. A simplified resonant global clock distribution with a single equivalent driver driving a large distributed clock network (modeled as a lumped RC element) is shown in Figure 1a. Although practical integrated implementations typically involve an array of distributed inductors, a single inductor is shown for simplicity.



(a) Simplified Schematic using lumped clock capacitance and a single inductor

(b) Clock voltage and inductor current waveforms

Figure 1: Basic resonant clock architecture

Figure 1b shows the observed voltage at clk and current flow I_L through the inductor. At frequencies close to the natural frequency f_0 the impedance of the resulting tank circuit increases, consuming lower lower power dissipation while sustaining the desired oscillation amplitude.

With resonant clocking, the per-cycle energy dissipation (E_{PC}) incurred in driving oscillations on a load capacitance C_{load} with peak-to-peak amplitude V_{dd} at frequency f_0 can be expressed as [13]:

$$E_{PC} = \frac{\pi}{4Q} C_{load} V_{dd}^2 \,, \qquad (1)$$

where Q is the system quality factor. Improving Q through low-loss inductors and clock networks reduces clock power dissipation in comparison to conventional clocking. At near-resonance frequencies, E_{PC} remains low as shown in Figure 1a, whereas it increases considerably away from f_0. Furthermore, as reported in [1], operating at frequencies sufficiently below f_0 warps the clock, compromising functionality. For this reason, current resonant clock implementations [10, 12] employ mode switches to disconnect the inductors, using conventional clocking at frequencies further away from f_0. Careful selection of f_0 is required to optimize the system for either Thermal Design Power (TDP) limited peak performance (higher f_0), or average battery life (lower f_0).

Dynamic Voltage and Frequency Scaling (DVFS) is a widely-used runtime technique for energy-efficiency in digital systems [5,

ISLPED'14, August 11–13, 2014, La Jolla, CA, USA.
Copyright 2014 ACM 978-1-4503-2975-0/14/08 ...$15.00.
http://dx.doi.org/10.1145/2627369.2627627.

8,9]. Depending on workload, performance and battery-life considerations, the system is tuned at runtime to operate over a wide range of voltage-frequency settings. In such systems, resonant clocking is only effective for a fraction of the operating time of the system, when the frequency is close to f_0. A wide-voltage-frequency range resonant clock architecture which will afford resonant-clocking efficiencies across the entire operating voltage-frequency range is therefore highly desirable.

More recently, an intermittent resonant clocking technique that generates a "blip" waveform, has been proposed [4]. Another approach involves using multiple parallel inductors, each connected by a series switch, to tune f_0 [10]. These methods are however, either limited in achievable frequency range, or are not voltage scalable due to reliability concerns driven by design and clock generator topology.

This paper proposes quasi-resonant clocking (QRC), a novel resonant clock architecture which demonstrates *true* voltage-frequency scalability while meeting several key requirements:

1. No adverse reliability impact: All circuit nodes remain within the supply voltage rails.
2. On-the-fly frequency and voltage scalability.
3. Energy-efficient operation over a wide, *continuous* tuning range from 0 to f_0.
4. Readily controllable clk duty-cycle.

The key observation enabling the proposed architecture is that resonant clock operation can (with the correct circuit topology) be periodically "suspended" for an arbitrary duration of time while the clock is held at V_{dd} or V_{ss}, allowing for true frequency scalability. Runtime-control plays a central role in enabling this scalability. The efficacy of QRC is demonstrated with post-layout simulations of the clock distribution network of a 64-bit pipelined Multiply-Accumulate(MAC) design.

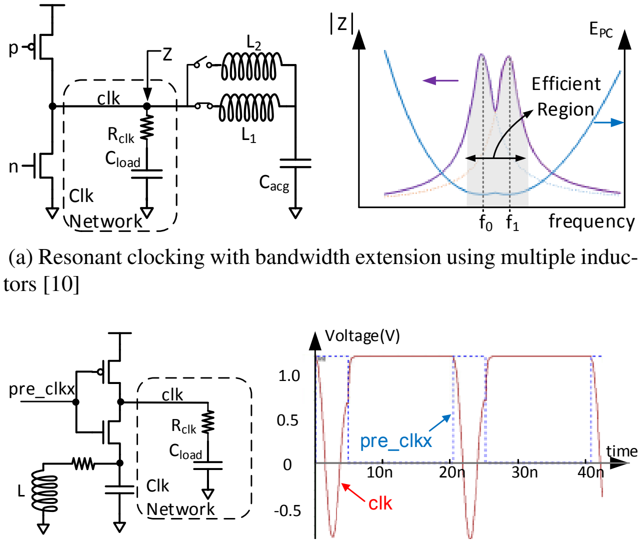
The rest of this paper is organized as follows. Section 2 provides an overview of the related work in the field. Section 3 presents the QRC architecture, including the clock generator topology and the control module which ensures robust DVFS support. Simulation results of the extracted 64-bit MAC unit clock network are discussed in Section 4, including voltage-frequency scaling, duty-cycle control, and energy dissipation.

2. RELATED WORK

Figure 2a illustrates a recently proposed solution involving the use of separate shunt inductors in parallel with a mode switch to make discrete f_0 adjustments [10]. Selecting the appropriate inductor combination enables the system to operate at the most efficient resonant configuration, extending the effective bandwidth.

While well-suited to its specific application and fabrication technology, the approach has limited frequency range, insufficient for wide-tuning range DVFS systems. Furthermore, placing extra inductors under already challenging placement constraints is not feasible in most commercial process technologies.

Another approach, illustrated in Figure 2b involves an intermittent resonant clock generator which creates a frequency-tunable "blip" [4] to enable frequency-scaling. This system has a number of limitations, however. By construction, the clock voltage transitions significantly outside the CMOS voltage rail. In addition to added power dissipation, the near-2V_{dd} swing of the clock is a reliability concern due to oxide-stress in connected MOS devices, limiting the feasibility to near-threshold voltage systems. Finally, the system does not allow for duty-cycle control, a key post-silicon test-optimization.



(a) Resonant clocking with bandwidth extension using multiple inductors [10]

(b) Intermittent clocking generating blip waveform [4]

Figure 2: Existing frequency-scalable resonant clock implementations

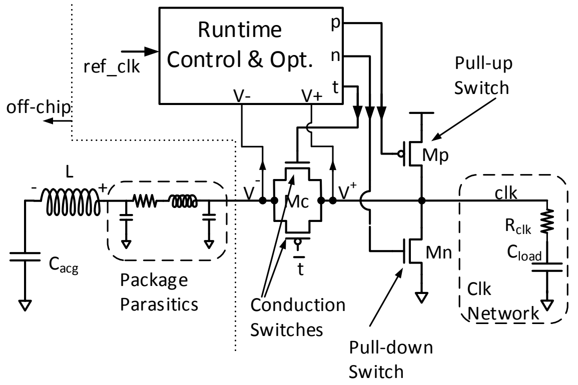
3. PROPOSED ARCHITECTURE

The central idea behind the proposed system stems from the observation in Figure 1b that as the clock reaches V_{dd} or V_{ss}, the current flow through the inductor, I_L equals 0. If L is disconnected from the network at this time, clk could be held to the appropriate supply rail indefinitely. Resonant clocking operation can be subsequently resumed by reconnecting the inductor to the clock distribution.

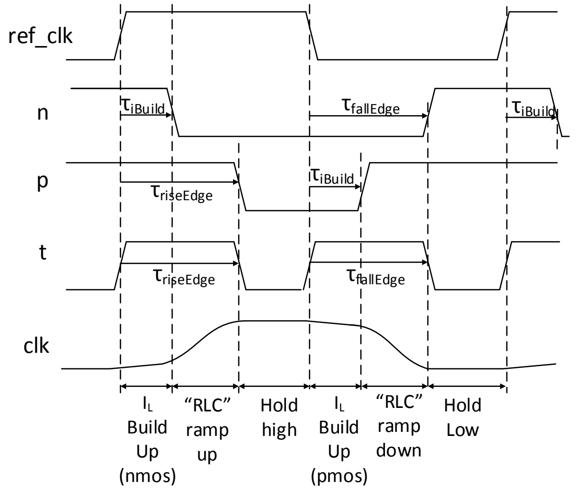
By relying on efficient LC resonance for clock transition, and conventional drive to retain clock-state at either V_{dd} or V_{ss}, QRC achieves robust and efficient wide-ranging frequency scalability, extending resonant clocking benefits across the entire frequency range of DVFS systems.

Figure 3 shows a simplified quasi-resonant clock implementation and timing diagram. The implementation discussed in this paper involves the use of an off-chip inductor. The resulting package parasitics shown in the figure are therefore included in all simulations. A conduction switch M_c, is added to the resonant clock architecture to disconnect the inductor from the clock network. A run-time control unit provides timing for n, p and t by evaluating the potential across conduction switch to determine current flow.

Consider the steady-state operation of the QRC system. The voltage across C_{acg} ($C_{acg} >> C_{load}$) is nearly steady at $Vdd/2$ throughout (for a 50% duty-cycle clock). At the start of the clock cycle, $V(ref\_clk) = 0$, $V(clk) = 0$. As ref_clk transitions to V_{dd}, t is asserted and connects the inductor to the resonant system, resulting in an RL current build-up in the inductor. After a duration τ_{iBuild}, n transitions to 0, turning off the hitherto conducting M_n. This begins the LC driven transition of clk toward V_{dd}. After a delay of $\tau_{riseEdge}$, clk arrives at its peak voltage, the control module senses the clk peak and de-asserts t, disconnecting the inductor from the network, and drives p to 0, turning on M_p and bringing clk to the V_{dd} rail. At the end of this sequence, clk has transitioned to V_{dd}, the inductor remains disconnected from the grid, enabling M_p to hold clk to V_{dd} indefinitely until the next transition of clk toward 0. The sequence of events orchestrating the falling clk edge is conceptually similar to the rising edge sequence and is illustrated in Figure 3b. The detection of the time instant of clk maxima or



(a) Simplified schematic



(b) Timing diagram of the quasi-resonant clock system

Figure 3: Proposed QRC architecture

minima coinciding with $I_L = 0$, is performed by comparing the potential difference across M_c.

Quasi-resonant clocking simulation waveforms are shown in Figure 4. The clock transitions between alternate resonant transition phases and conventional hold phases. It is noteworthy that even with perfect timing, I_L does not remain 0 after M_c is off (during the $conv.phase$). The parasitic capacitance in the package trace at V^-, which is at one of the supply rail voltages at the time of disconnection, experiences an under-damped LC oscillation. Ensuring that this oscillation remains contained within the supply rails and does not adversely affect reliability requires accurate timing control of t.

The *continuous* range of frequencies that can be obtained by QRC is $0 < f \leq f_{max}$, where:

$$f_{max} \approx \frac{1}{\sqrt{LC_{load}}}, \tag{2}$$

Considerations that govern the slew rates of QRC waveforms are identical to those of regular resonant clocks. These slew rates will be lower than conventional clocks [1,4,10,12,16]. Degraded slews impact efficiency and performance due to increased crowbar current and process-dependent clock skew respectively. These challenges can be effectively addressed using a variety of circuit and architectural approaches [7, 12, 16], and are outside the scope of this work.

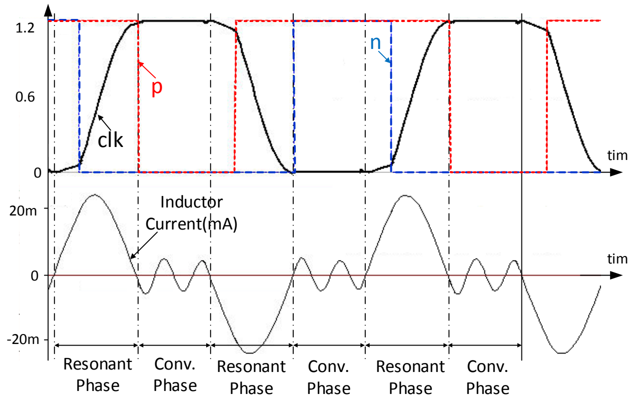


Figure 4: Quasi-resonant clocking simulation waveforms

3.1 Dual-Delay-Locked Loop

Timing requirements for n, p and t relative to ref_clk pose a number of design challenges. It is crucial that t be de-asserted at the exact time that $I_L = 0$, regardless of current-sense delay, or pre-driver latency. Furthermore, the timing for n and p differ for rising and falling transitions of clk. Notice from Figure 3b that while the timing of n (p) during the clk rise (fall) is driven by current build-up requirements, it is the clock transition duration (determined mainly by L and C_{load}) which governs timing during the fall (rise) of clk. Achieving a zero-delay signal-path between the current-sensing mechanism, and t, and enabling context-specific timing requirements for the n and p signals was best achieved by devising a Dual-Delay-Locked-Loop (Dual-DLL).

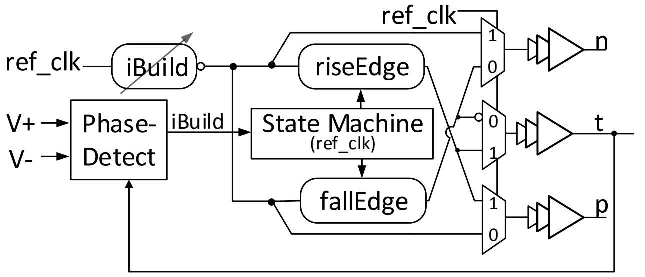


Figure 5: Proposed Dual-DLL for context-specific timing control

Figure 5 shows the Dual DLL module which performs runtime control and optimization of the M_c, M_n and M_p switches. The $iBuild$ delay chain determines the current-buildup duration in the inductor before launching a clock transition. The $riseEdge$ and $fallEdge$ delay chains control the time duration from the de-assertion of M_n (M_p) and the assertion of M_p (M_n) . As depicted in Figure 3b, this duration corresponds to the RLC transient driving the clock to V_{dd} (V_{ss}).

The phase detector senses I_L polarity using V^+ and V^-, and provides the dual-DLL with an $iSink$ signal, indicating whether the inductor is sourcing current to, or sinking current from clk. This feedback allows the DLL to adjust $riseEdge$ ($fallEdge$) delays, enabling de-assertion of t when $I_L = 0$ independent of pre-driver latency. Consequently, it is crucial that the phase detector sampling edge and the asserting/de-asserting edge of M_c be driven by the same physical signal.

Given the "double-pumped" nature of the phase-detector, with two evaluations per-cycle (peak and valley detection), $riseEdge$ ($fallEdge$) delay adjustments are made during the falling (rising) edge of the ref_clk to ensure glitch-less n, p and t signals.

Multiplexers at the end of the control path provide the necessary "context-switching" for n, p and t between rising and falling clk edges. The $iBuild$ module is currently not controlled at run-time in the current implementation, resulting in an opportunity loss for further energy optimization.

During an initial power-up or reset sequence, the dual-DLL will not be in lock, resulting in mis-timed inductor disconnection and beyond supply-rail V^- oscillations. To address this issue, the control module "warms-up" to the final state by starting with only a sub-bank of M_c switches, interleaving delay-locking with the additional MC banks until lock is achieved with all M_c banks on. Subsequently, the control module goes into a low-bandwidth (low-power) thermal-tracking mode, and reverts to high-frequency relock only during a voltage or frequency change event.

3.2 Phase-detector

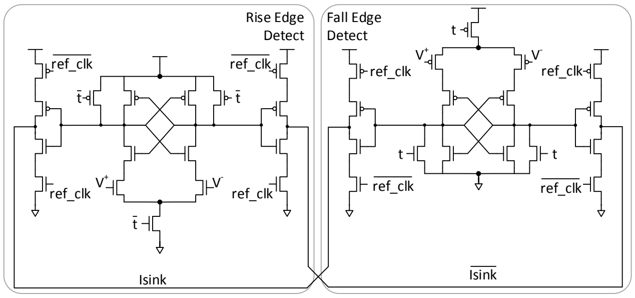


Figure 6: Phase detector used in Dual-DLL implementation

The phase-detector used in the control module is shown in Figure 6. The voltages V^+ and V^-, are both close to either V_{dd} and V_{ss} when sampled to determine I_L direction. Resolving differences at such extreme common-mode voltages with a single latch would be ineffective. Consequently, a dual-sense-amp latch structure was developed. Phase detection is always triggered by a de-assertion of t. Toward the end of a clk rise (fall) transition, if $V^+ > V^-$ at the sampling instant, then the inductor is sinking current and t needs to be de-asserted sooner (later).

The voltages of V^+ and V^- remain close to V_{dd} or V_{ss} and are sampled by the Dual-DLL control before the evaluating phase of the phase-detector resets, enabling the omission of keepers and the traditionally-employed connection device between the differential and cross-coupled inverter pairs.

To address metastability, the phase detector output is double-latched by both rising and falling edge flops, and used by the appropriate Dual-DLL control logic sections to affect $fallEdge$ and $riseEdge$ respectively.

4. EXPERIMENTAL RESULTS

To validate the operation and efficiency of the QRC architecture, a pipelined 64-bit Multiply-ACcumulate (MAC) unit was implemented using commercial Synthesis, Auto Place and Route (SAPR) tools in an industrial 65nm CMOS process. All simulations include post-layout parasitics from the MAC design, back-annotated parasitics for the drivers, conduction switches.

4.1 Setup

Three separate MAC variants were implemented to accurately quantify the benefits of the proposed clock architecture.

4.1.1 Quasi-resonant clock (res_clk)

An industrial place-and-route tool was augmented with a parameterizable resonant clock tree and mesh generator to build the quasi-resonant distribution. The remainder of the design was implemented using a conventional place-and-route flow. Package trace RLC parasitics leading from the conduction switch to the off-chip inductor were included in all simulations, as was power dissipation of the control module, the pre-driver leading up the final resonant clock driver. Following the completion of the MAC design, the clock distribution network was extracted and used for analysis of power. Simulation measurements of the quasi-resonant distribution indicate a clock skew of 6.1ps when V_{dd}=1.2V. Consistent with expectations, the skew remains unchanged at 0.8V.

The QRC controller was implemented using a conventional SAPR flow and its power dissipation is included in the total power dissipation reported.

4.1.2 Conventionally-driven resonant clock distribution (res_conv_clk)

This variant of the design uses the resonant clock tree network but employs *conventional* drive to achieve the same clock skew and slew targets (The inductor, decap and conduction switch are omitted). This variant serves only as a baseline to evaluate the efficiency of res_clk over CV^2 dissipation while driving the same load. Consequently, the significant electromigration (SigEM) challenges at M_n and M_p, resulting from the removal of the inductor are ignored.

4.1.3 Conventional clock tree ($conv\_clk$)

To enable a fair assessment of the efficiencies of quasi-resonant clocking efficiency, a conventional clock distribution was implemented using the SAPR flow with a relaxed skew target of 70ps, and a slew target similar to that of the resonant clock distribution. As in the res_conv_clk implementation, SigEM challenges in this implementation are ignored.

The clock tree synthesis tool did not take advantage of the entire slew budget. The consequent sharper edges in the conventional design would result in lower post-silicon variation-driven skew. Monte-Carlo simulations were therefore performed to quantify the variation-driven clock skew benefit of the conventional clock over res_clk. A 3σ skew-credit was then provided, resulting in a relaxed skew target of 70ps.

4.2 Frequency-only Scaling

Figure 7 demonstrates the capability of the QRC system to perform on-the-fly frequency-scaling. Clock waveforms corresponding to frequency points a, b and c on the frequency-time plot are overlaid. For larger time-periods, the controller need make no adjustments to the delay-chains as frequency scales, and remains locked since the current build-up and transition times are independent of clock frequency – Directly scaling ref_clk achieves the desired scaling result. At lower cycle-times, the under-damped response at V^- following the de-assertion of t has a minor timing impact on clk node maximum and minimum, requiring some delay-chain setting adjustment.

4.3 Voltage-Frequency Scaling

Unlike QRC frequency-scaling, which is relatively straightforward, voltage-scaling is more challenging since voltage-scaling varies delay-chain latencies that need to be maintained. To demonstrate DVFS support, the ref_clk was sourced from a ring-oscillator in the same voltage domain to provide the necessary accompanying change in frequency.

DVFS simulation results of the clock are shown in Figure 8. The initial voltage of 1.3V was scaled down to 0.8V continuously with an accompanying frequency reduction. The runtime control mod-

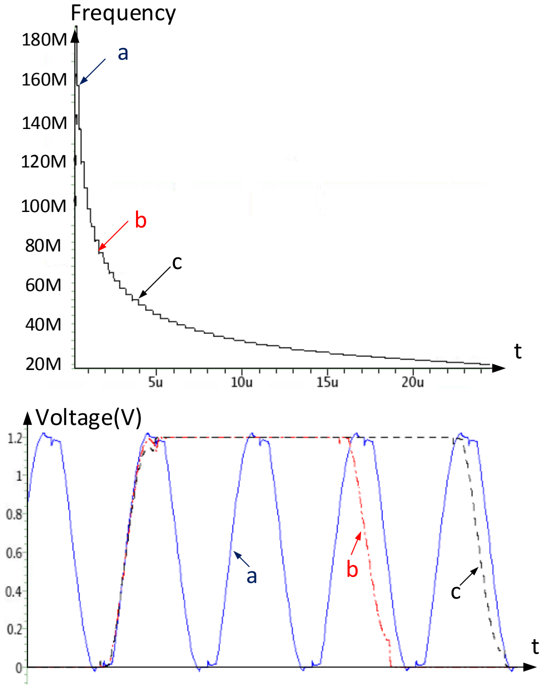


Figure 7: On-the-fly wide-range frequency scaling resonant clocking: clk simulation waveforms

ule bandwidth is sufficient to maintain lock. Clock waveforms corresponding to various voltage-frequency combinations are overlaid and demonstrate correct QRC operation.

4.4 Duty-cycle Control

Duty-cycle control is an important requirement in most clock designs. Dynamic-logic, level-sensitive latches, or other phase-paths require clock duty-cycle control for robust operation. Duty-cycle tuning is also a common post-silicon yield and performance optimization. To the best of the author's knowledge, QRC is the only wide-frequency range resonant clock architecture capable of programmable duty-cycle support. Figure 9 shows the duty-cycle ratio (DCR) of clk as ref_clk DCR varies, demonstrating a wide duty-cycle tuning range.

4.5 Decoupled Efficiency-Frequency behavior

To illustrate the key difference between existing resonant designs and the proposed QRC architecture, constant-voltage frequency scaling in the 2MHz–200MHz range is conducted while monitoring the active E_{PC}.

Figure 10 illustrates the *active* (no leakage) E_{PC} of the quasi-resonant clock and a baseline corresponding to the E_{PC} incurred in driving the same capacitance conventionally (res_conv_clk). QRC energy-efficiency remains nearly constant over 2 decades of frequency-scaling, resulting in approximately 60% savings compared to res_conv_clk . This is in stark contrast to the narrow frequency range of efficient resonant clock operation in current designs [1, 10, 12], even at a constant voltage. QRC always operates "at resonance" from an energy dissipation perspective, regardless of the operating frequency.

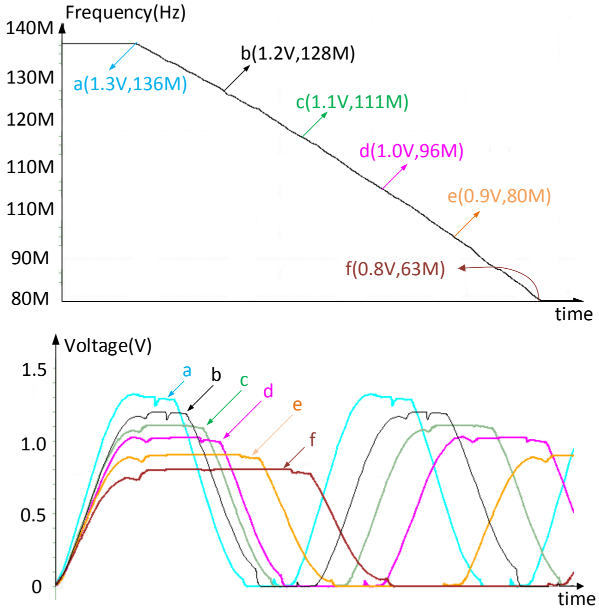


Figure 8: On-the-fly wide-range voltage-frequency scaling resonant clocking: clk simulation waveforms

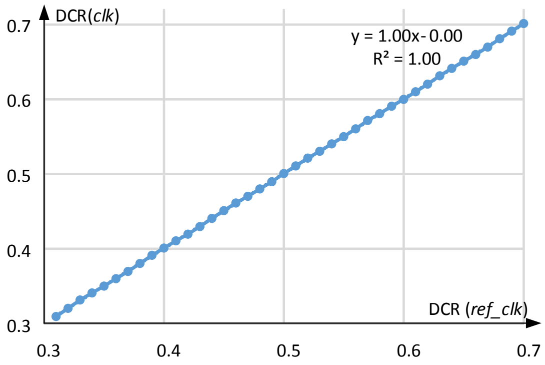


Figure 9: QRC simulation demonstrating ref_clk DCR vs clk DCR

One interesting observation from Figure 10 is the fluctuation in E_{PC} of the quasi-resonant distribution at high frequencies. This is caused by the voltage oscillations at node V^- (infer from Figure 4) after the inductor is disconnected. Depending on the fractional relationship between load capacitance and parasitic capacitance on V^-, I_L can either be constructive or destructive at the onset of resonant transition. This interference gradually diminishes at lower frequencies as the V^- oscillations die out and $I_L = 0$ at the onset of the next resonant transition.

4.6 Energy Efficiency

The energy efficiency of the three design variants, res_clk, res_conv_clk and $conv\_clk$ were evaluated. Starting with a 1.3V supply at 200MHz, supply voltage and frequency were scaled following a DVFS profile down to V_{min} at 0.8V. All measurements are made at $25\,^{\circ}$C

Figure 11 shows total E_{PC} dissipation of the three variants across the voltage-frequency scaling range. In this experiment, res_clk efficiency relative to res_conv_clk and $conv\_clk$ vary with frequency because voltage scaling increases the resistance of the M_n, M_p and M_c devices (Figure 3a). Maintaining efficient operation at a lower voltage requires drive strength modulation of drivers [13] or voltage-boosting techniques [4] which have not been implemented in the current work. Nevertheless, throughout the entire voltage range, res_clk dissipates at least 54% lower energy per cycle then

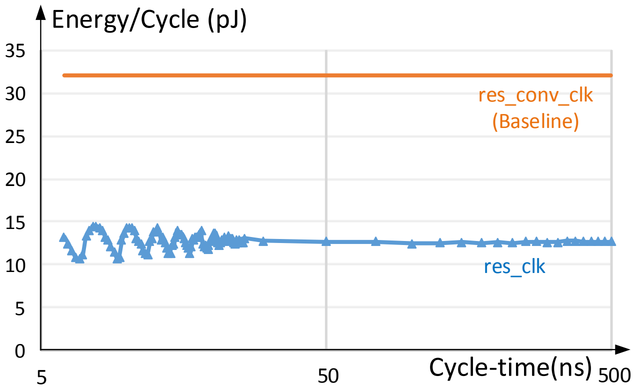


Figure 10: E_{PC} vs. cycle-time

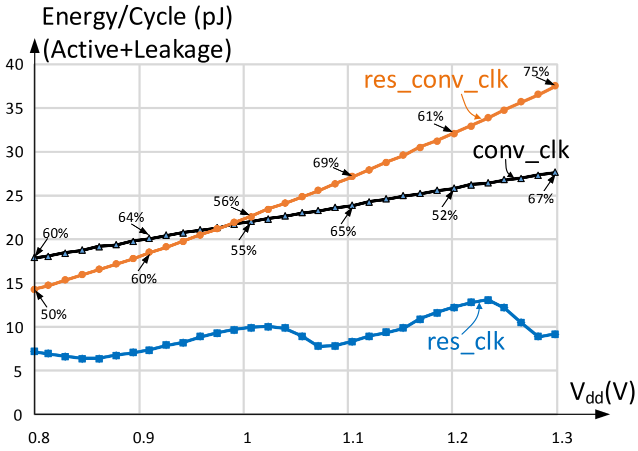


Figure 11: E_{PC} vs. V_{dd}. QRC savings annotated.

either res_conv_clk or $conv\_clk$. E_{PC} fluctuation for res_clk is expected as explained in section 4.5. The resonant clock distribution has a higher total clock load compared to a conventional clock tree due to the wider metal wires and the use of a clock grid. Consequently at higher voltages, $conv\_res\_clk$ dissipates more power than $conv\_clk$. As voltage scales however, energy dissipation in $conv\_clk$ reduces less rapidly due to the increased leakage in the clock buffers, resulting in comparable E_{PC} at 1.0V.

5. CONCLUSION

Quasi-resonant clocking (QRC) is presented in this paper. The proposed solution for the first time, enables true voltage-frequency scalability, allowing resonant clocking efficiencies to be effectively harnessed along with Dynamic Voltage and Frequency Scaling (DVFS) techniques. QRC enables a continuous operating frequency range of $0^+ - f_0$. Simulations of a pipelined 64-bit Multiply-Accumulate clock distribution in a commercial 65nm CMOS process are presented to demonstrate on-the-fly voltage-frequency scaling from 0.8V–1.3V. Energy savings of 50%-75% over a conventional clock distribution are demonstrated. QRC simulations over a frequency range of 2MHz-200MHz are also presented.

6. ACKNOWLEDGEMENTS

The author thanks Sanjay Pant and Radha Poovendran for helpful discussions.

7. REFERENCES

[1] S. Chan et al. A Resonant Global Clock Distribution for the Cell Broadband Engine Processor. *Journal of Solid State Circuits*, 44(1):64–72, Jan. 2009.

[2] S. Chan, P. Restle, and K. Shepard. A 4.6GHz resonant global clock distribution network. In *International Solid State Circuits Conference*, Feb. 2004.

[3] A. J. Drake et al. Resonant Clocking using Distributed Parasitic Capacitance. *Journal of Solid State Circuits*, 39:1520–1528, Sep. 2004.

[4] H. Fuketa, M. Nomura, M. Takamiya, and T. Sakurai. Intermittent resonant clocking enabling power reduction at any clock frequency for 0.37v 980kHz near-threshold logic circuits. In *International Solid State Circuits Conference*, pages 437–437, Feb. 2013.

[5] K. Gillespie et al. Steamroller: An x86-64 Core Implemented in 28nm bulk CMOS. In *International Solid State Circuits Conference*, Feb. 2014.

[6] M. Hansson, B. Mesgarzadeh, and A. Alvandpour. 1.56GHz On-Chip Resonant Clocking in 130nm CMOS. In *Custom Integrated Circuits Conference*, pages 241–244, Sep. 2006.

[7] D. Jeon et al. A Super-Pipelined Energy Efficient Subthreshold 240MS/s FFT Core in 65nm CMOS. *IEEE Journal of Solid-State Circuits*, 47(1):23–34, Jan. 2012.

[8] N. Kurd et al. Next generation intel core micro-architecture (nehalem) clocking. *IEEE Journal of Solid-State Circuits*, 44(4):1121–1129, Apr. 2009.

[9] H. McIntyre et al. Design of the Two-Core x86-64 AMD Bulldozer Module in 32nm SOI CMOS. *IEEE Journal of Solid-State Circuits*, 47(1):164–176, Jan. 2012.

[10] P. Restle et al. Wide-Frequency-Range Resonant Clock with On-the-Fly Mode Changing for the POWER8 Microprocessor. In *International Solid State Circuits Conference*, pages 100–101, Feb. 2014.

[11] P. J. Restle et al. A Clock Distribution Network for Microprocessors. *Journal of Solid State Circuits*, 36:792–799, May. 2001.

[12] V. Sathe et al. Resonant-clock Design for a Power-efficient, High-volume x86-64 Microprocessor. In *International Solid State Circuits Conference*, pages 68–69, Feb. 2012.

[13] V. Sathe et al. Resonant-Clock Design for a Power-Efficient, High-Volume x86-64 Microprocessor. *IEEE Journal of Solid-State Circuits*, 48(1):140–149, Jan. 2013.

[14] V. S. Sathe, J. Y. Chueh, and M. C. Papaefthymiou. Energy-Efficient GHz-Class Charge Recovery Logic. *Journal of Solid State Circuits*, 42(1):38–47, Jan. 2007.

[15] V. S. Sathe, J. C. Kao, and M. C. Papaefthymiou. RF2: A 1GHz FIR Filter with Distributed Resonant Clock Generator. In *IEEE Symposium on VLSI Circuits*, pages 44–45, Jun. 2007.

[16] V. S. Sathe, J. C. Kao, and M. C. Papaefthymiou. Resonant-Clock Latch-Based Design. *Journal of Solid State Circuits*, 43(4):864–873, Apr. 2008.

[17] G. Shamanna, N. Kurd, J. Douglas, and M. Morrise. Scalable, sub-1w, sub-10ps Clock Skew, Global Clock Distribution Architecture for Intel Core i7/i5/i3 Microprocessors. In *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*, pages 83–84, June 2010.

[18] J. Wood, T. Edwards, and S. Lipa. Rotary Traveling-wave Oscillator Arrays: a New Clock Technology. *Journal of Solid State Circuits*, 36:1654–1665, Nov. 2001.