5-5-2015

Power Reductions with Energy Recovery Using Resonant Topologies

Ignatius S.A. Bezzam
Santa Clara University

SANTA CLARA UNIVERSITY

Department of Electrical Engineering

Date: May 5, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Ignatius Samuel Augustine Bezzam

ENTITLED

POWER REDUCTIONS WITH ENERGY RECOVERY USING RESONANT TOPOLOGIES

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

Thesis Advisor	Thesis Reader
Chairman of Department	Thesis Reader
	Thesis Reader
	Thesis Reader

POWER REDUCTIONS WITH ENERGY RECOVERY USING RESONANT TOPOLOGIES

by

Ignatius S. A. Bezzam

Dissertation

Submitted in partial fulfillment of the requirements for

the Degree of Doctor of Philosophy

in Electrical Engineering

in the school of Engineering

at Santa Clara University, May 5, 2015

Santa Clara, California USA

Dedicated to

Laura Maria Carcione

my better half and far more than what Petrarca imagined.

ACKNOWLEDGEMENTS

"If I love, what business is it of yours?" – Johann Von Goethe

This whole work has been about doing what I love, even if it is nobody's profit making business; something I could not do even in my childhood, but got to finally in my 50's. I am passionate about the electrical engineering fields of Analog/RF, High Speed Digital and Power management. This thesis is a statement of it. It is dedicated to my better half as that is the least I can do for all the love and support I received like nobody's business.

I want to sincerely thank my advisor Dr. Shoba Krishnan, professor in the Department of Electrical Engineering, Santa Clara University in supporting me with the freedom to disagree and that too with heart and resources. In addition, her ability to provide a clear cut direction in the presentation of data has helped me tremendously in my written communication success so far. So if this dissertation is indeed readable, the credit goes to her above 90%.

I am also indebted to Prof. C. Mathiazhagan who has spurred, inspired and challenged me since we did undergrad in IIT Madras in 1993. I thank him immensely for taking the time to talk regularly, guide and encourage me to work towards the completion of this work. His brilliance, patience, friendship and attention to detail have helped me cross many barriers in this work.

I would also like to thank my "low power guru" Dr. Tezaswi Raja for agreeing to be on the Ph.D. committee and co-author several papers with me, in midst of all his silicon and carbon tape-outs. I am indebted to Prof. Samiha Mourad for directly guiding my research and fruitful interactions with other doctoral students. I am also thankful to Prof. Tokunbo Ogunfunmi for the intensely useful courses and continuous support of my program. I am lucky to have had thought provoking, delicious dialogs

with Dr. Ahmed Amer, on my Ph.D. committee. Once again, I thank my advisor and all the committee members for taking the time to decide on my courses, teach and keep track of the progress of this work. I am grateful to all my other instructors, from SCU faculty and outside, for various courses and probing questions answered.

Last but not the least; I am proudly grateful to my son Eric Francis, who is completing his Electrical & Computer Engineering Bachelor of Science in Germany, for challenging me to give him more to follow and for competing to finish his doctorate before mine!

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Power reductions with energy recovery using resonant topologies

Ignatius Bezzam
Department of Electrical Engineering
Santa Clara University
Santa Clara, California

2015

ABSTRACT

The problem of power densities in system-on-chips (SoCs) and processors has become more exacerbated recently, resulting in high cooling costs and reliability issues. One of the largest components of power consumption is the low skew clock distribution network (CDN), driving large load capacitance. This can consume as much as 70% of the total dynamic power that is lost as heat, needing elaborate sensing and cooling mechanisms. To mitigate this, resonant clocking has been utilized in several applications over the past decade. An improved energy recovering reconfigurable generalized series resonance (GSR) solution with all the critical support circuitry is developed in this work. This LC resonant clock driver is shown to save about 50% driver power (>40% overall), on a 22nm process node and has 50% less skew than a non-resonant driver at 2GHz. It can operate down to 0.2GHz to support other energy savings techniques like dynamic voltage and frequency scaling (DVFS).

As an example, GSR can be configured for the simpler pulse series resonance (PSR) operation to enable further power saving for double data rate (DDR) applications, by using de-skewing latches instead of flip-flop banks. A PSR based subsystem for 40% savings in clocking power with 40% driver active area reduction

is demonstrated. This new resonant driver generates tracking pulses at each transition of clock for dual edge operation across DVFS. PSR clocking is designed to drive explicit-pulsed latches with negative setup time. Simulations using 45nm IBM/PTM device and interconnect technology models, clocking 1024 flip-flops show the reductions, compared to non-resonant clocking. DVFS range from 2GHz/1.3V to 200MHz/0.5V is obtained. The PSR frequency is set >3× the clock rate, needing only $1/10^{th}$ the inductance of prior-art *LC* resonance schemes. The skew reductions are achieved without needing to increase the interconnect widths owing to negative set-up times.

Applications in data circuits are shown as well with a 90nm example. Parallel resonant and split-driver non-resonant configurations as well are derived from GSR. Tradeoffs in timing performance versus power, based on theoretical analysis, are compared for the first time and verified. This enables synthesis of an optimal topology for a given application from the GSR.

1 INTRODUCTION

There are fundamental electrical engineering principles underlying the severe problem of managing the power that produces heat dissipation in SoCs and processors operating at GHz clock rates. A literature survey of the current state-of-art on addressing this problem shows the limitations in various solutions available now. The energy usage from a top down and bottom up perspective is examined in order to understand the metrics to be maintained while power is reduced.

1.1 Motivation for Wide Frequency Energy Recovery

Laptops cannot be operated on top of laps anymore due to the intense heat generated. To solve the same issue on a larger scale, cooling costs in the order of $50billion/year are needed for just small businesses. Businesses use farms of workstations for computing which are made of ICs. As shown in Figure 1.1, these costs are quickly outpacing the cost of hardware due to the thermal costs associated with ICs consuming 100’s of watts of power. This is primarily because of these thermal problems from power densities of microchips. There is an increase in the power consumed per transistor as well as the number of transistors on a single IC die. Silicon chips using deep sub-micron (DSM) nanometer scale processors can now reach the temperature of a rocket nozzle. They may soon have spots as hot as the surface of the sun. To handle this and the consequent reliability concerns, elaborate sensing and thermal management are required. Thus, power consumption is a key issue in high performance systems based on processors (CPUs and GPUs) as they consume hundreds of watts as shown in Figure 1.1. Higher IC power results in increased energy bills for companies.

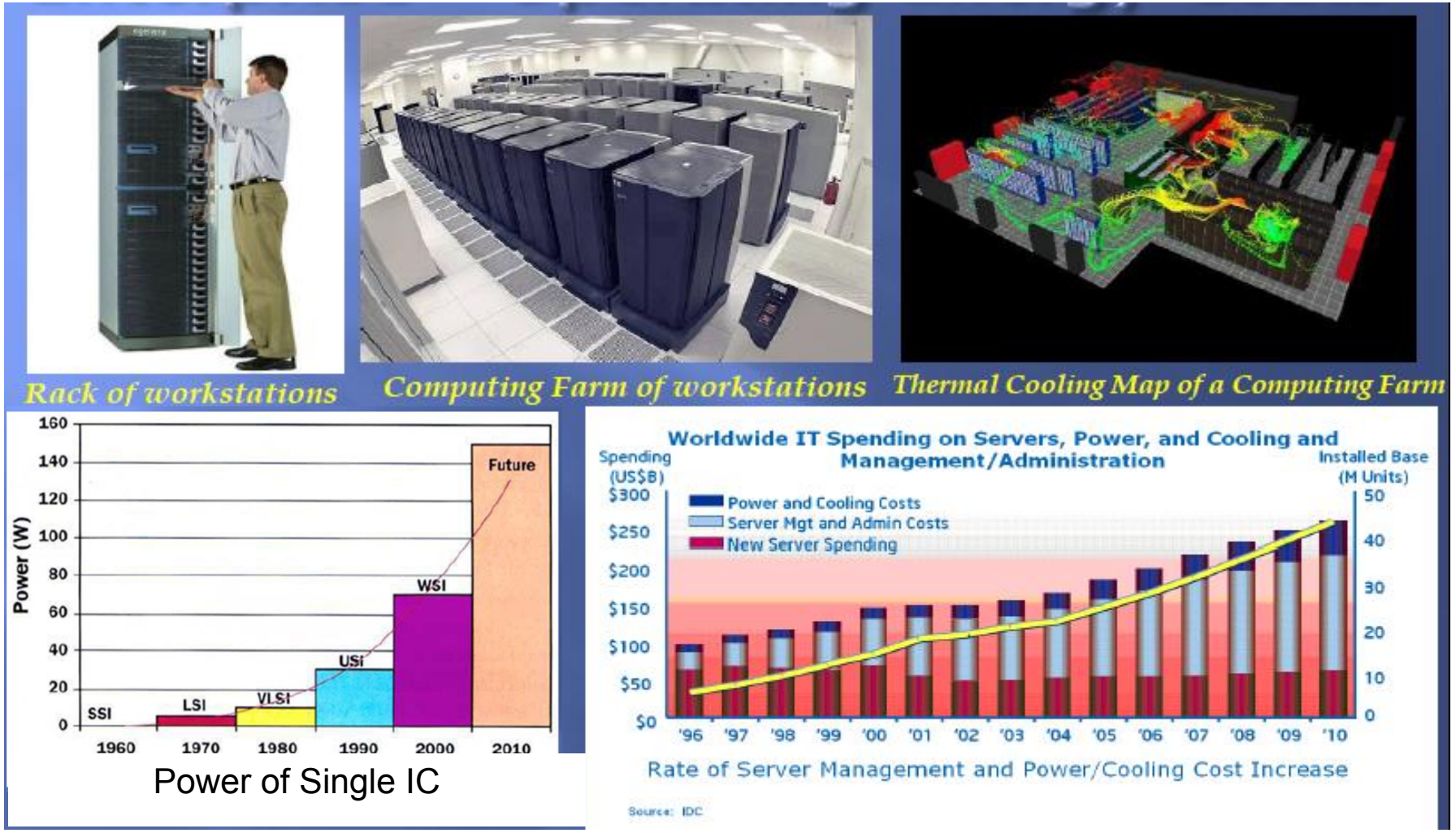


Figure 1.1 System expenses and root causes

(courtesy Dr. T. Raja, NVIDIA Corporation, Lecture on Low Power Design).

Thus, there is an urgent need for low power techniques for the following reasons,

a) Increase battery lifetime and/or decrease number of solar cells
b) Reduce Cooling Fixtures, Form factor and Costs
c) Increase Reliability & Sustainability

VLSI circuits operating in GHz range typically have switching power dissipation much larger than leakage losses. For example, high end GPUs can take over 300Watts. To meet stringent skew requirements (<8ps across 64mm^2 chip from AMD shown in [11]), synchronous clocking alone can take 24%-70% of power from processors to SoCs.

In clock power reduction, DVFS is a very important technique in runtime power management, and is extensively used by high performance processors. Here part or all of the IC is dynamically scaled to run at the minimum frequency needed and the supply voltage scaled to the minimum needed to support the minimum frequency. All other energy recovery techniques need to incorporate wide frequency operation supporting DVFS. Prior-art resonant solutions inherently do not do that.

Recovering continuous switching energy from clocking that is spread all over the chip not only saves power but can also eliminate cooling costs. An all-important performance metric to be maintained while achieving power reduction is the timing closure that involves a host of specifications like skew, jitter, delay variations etc. Some of the resonant schemes deteriorate these while achieving power savings and this may not be acceptable. So an additional requirement on any new resonant solution is to achieve lower skew while reducing power, especially at higher frequencies.

Thus, the aim of this work is to arrive at energy recovering resonant solutions that inherently operate over wide frequencies and give better

performance in terms of lower skew and jitter for timing closure. This dissertation examines solutions to various limitations in using prior-art resonant clock drivers and the best way to use their energy recycling feature over a wide frequency range. A novel reconfigurable scheme called generalized series resonance (GSR) is proposed. This can be dynamically programmed into various series or parallel resonance modes of operation for optimal trade-off. Closed-form design equations determining the power consumption improvements are arrived at while analyzing the timing performance at the clock sink points to enable automatic design synthesis.

Special flip-flops for ultra-low energy applications usually need to be designed to work with low amplitude signals from global clock grids from resonant clocks. The design of these is described here for applications where it is demonstrated to save further power. It is also desirable to have the new resonant schemes be able to directly drive standard flip-flops and gates to fit the standard design flow. Resonant techniques that can be used in the data path are also desirable to recover more energy, over and above the clock power reductions.

The new *LC* resonance operation proposed in this work is engaged only for the rise and fall transitions, rather than the entire clock period, and thus is not tied to one clock frequency. Energy recovery is then achieved over a much wider frequency range enabling DVFS. Run time optimization of the operation, through pulse width control, results in more savings of the clock power. CDNs savings can total to several watts of power in current DSM processors, SoCs and ASICs.

1.2 Literature review

Power dissipation considerations continue to dictate the use of multi-core architecture in processors and SoCs in technologies beyond 45nm [1], [2]. A full chip clock distribution network (CDN), meeting stringent timing requirements, can alone

take 25% of total power in processors and sometimes as much as 70% in SoCs [3]. Transistor scaling using ‘More of Moore’s law’ reduces area and gives faster transistors. Power densities are significantly higher when the constant voltage scaling method is used [4]. Due to the cooling costs needed to contain the large power densities, there has been an abrupt halt in the clock frequency increase even though the transistors themselves can switch much faster [5]. This calls for improvements beyond Moore’s law scaling.

The so called ‘More than Moore’ solutions [6] can be applied for this dilemma for reducing power using MEMS/NEMS resonators [7]. These technologies are not main-stream yet and involve additional costs. Architectural choices, like the use of multi-cores, give higher throughput using lower clock frequencies, resulting in lower power densities [4].

A low cost way is to use the passive components like metal spiral inductors, already available on standard process technologies [8], to consume less power in switching. Even in multi-core processors, total energy can be further reduced by using inductors. The energy used to charge the clock grid node capacitance (C) each period can be recovered and reused with an integrated inductor (L) in parallel, forming a resonant tank network [3]. The recovered energy would have been otherwise dissipated as heat. *LC* resonant circuit operation for reducing power consumption in high speed clocking applications has been extensively reported [11]–[14]. Since only losses need to be overcome at resonance, after the initial start-up, additional power savings can be realized by reducing the strength of clock buffers driving the *LC* load. Such recovery techniques are currently used in nanometer commercial processors for global clocking [11]. Even in multi-core processors, total energy can be further reduced by using inductors. Fully integrated LC resonant clocking is emerging to be

commercially viable on standard CMOS technology for reducing power consumption in the Clock Distribution Network (CDN) by energy recovery and reuse [1]. These continuously parallel resonant (CPR) solutions save 25% power or more, albeit over a narrow range of clock speeds.

Integrated inductor based tuned circuits have long been used for efficient power transfer in small signal radio frequency (RF) amplifiers [15]. They are extensively used in integrated DC-DC converters at large voltages and currents, albeit at low frequencies [16], [17]. These inductors are now well characterized for commercial use. Their use for clocking presents unique challenges, as operation with large signals and at very high (GHz) frequencies is needed [3], [18].

In order to reduce power as much as possible, modern high performance mobile designs are also using increasing number of voltage domains and regional clock trees [18]. Thus, it is beneficial to extend resonant solutions from global to regional clocking shown in Figure 1.2 [19]-[22]. However, the smaller capacitance values from local trees will dictate larger values of inductances for the same LC resonance frequency [23].

Resonant clock solutions extending the operating frequency range for DVFS have been reported [1], [7], [23]- [26]. Parallel resonance structure, as described in Chapter 2, can switch in multiple inductors for different frequency ranges as shown in [1], [25].

Chapter 3 describes series resonance topology that inherently gives wide frequency operation [23], [24], [26]. Pulsed mode resonance described in [23], [24] uses special latches to achieve best savings of power and area. Series resonance driver scheme in [7] generates flat top outputs that could directly drive standard logic cells.

However, the supporting control signals need special circuits to generate them, which have not been published in detail. This thesis describes them in detail in Chapter 4.

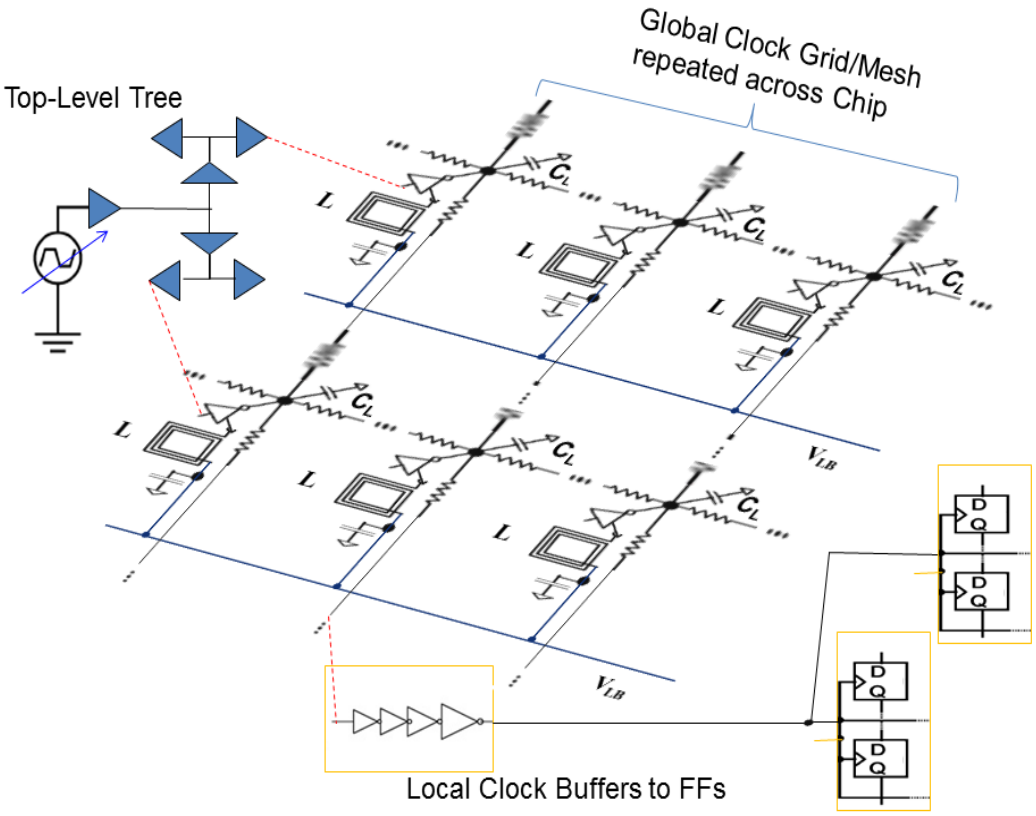


Figure 1.2 A Comprehensive Clock Distribution and Data Capture Scheme.

A silicon validation of a simplified series resonance called Intermittent Resonance (IR) is described in [24] and shows promising future for this work not yet realized in silicon.

1.3 Top-down Clock Distribution

Figure 1.2 shows the integration of resonant and non-resonant clock drivers at various levels of CDN, which will be treated in detail in later chapters. The numerous active and distributed passive components involved are detailed later. Figure 1.2 is an example of a grid based CDN. In synchronous SoC designs, 66% of clock power may

be dissipated in the local buffer stages driving the flip-flops [21]. From a high level perspective, for real life clocking applications in high speed computing and communication, timing closure is of utmost importance for functionality, performance, and yield [14], [18]. Lowering power at the expense of timing parameters like insertion delay variations, slew rates, skew and jitter may not be acceptable [3], [18].

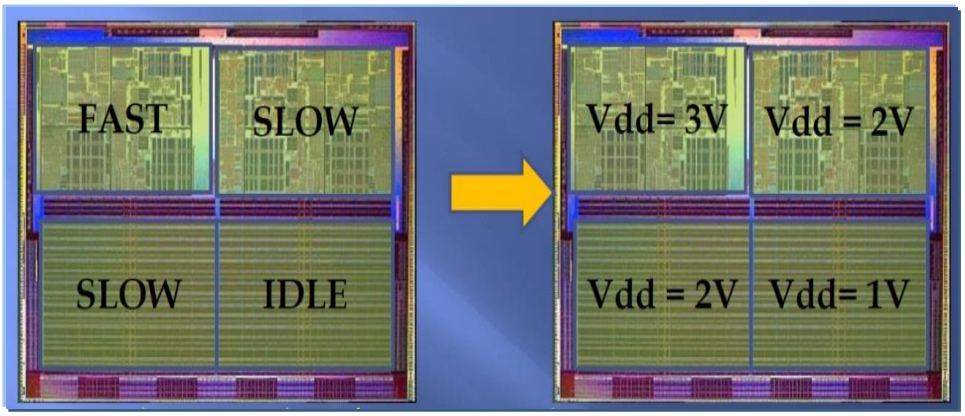


Figure 1.3 Dynamic Voltage and Frequency Scaling.
(courtesy Dr. T. Raja, NVIDIA Corporation, Lecture on Low Power Design)

Another important system level requirement is the ability to operate the same chip at different frequencies in different parts as shown in Figure 1.3. At a system level the strategy to minimize power is to operate only as fast as necessary and at the lowest voltage supporting that clock speed.

As will be seen, a sizable portion of the dynamic power is taken up by the clock distribution itself, to maintain the synchronous nature of the system.

1.4 Bottom-up View of Non-Resonant (NR) Digital Circuits

The root cause of power wasted in digital circuits and the reason for the runaway in thermal issues is now examined. As a baseline for power calculations and timing performance, equations for known drivers are considered first [27], [28]. Figure 1.4 shows a low power clock driver topology with no resonance (NR) driving a

large capacitive load C_L. Various parasitic resistors and lumped interconnect parasitics that can affect the slew rates and delays are shown. Switch parasitic capacitances are neglected compared to C_L. Output is near 50% duty cycle though input pulses are not.

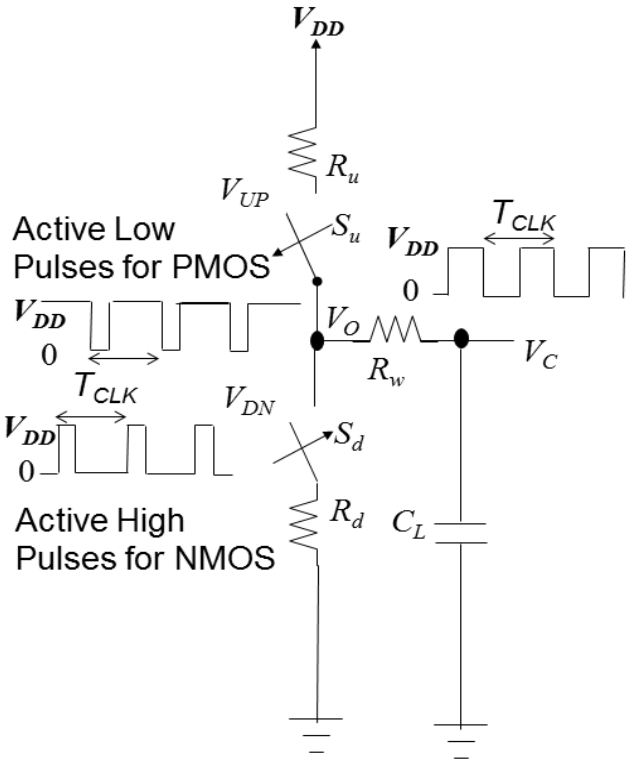


Figure 1.4 Clock Driver Topology for NR.

The split pull up and pull down scheme in Figure 1.4 minimizes the short circuit currents and thus consumes minimum dynamic power [29], [30]. This is at the expense of more circuit area, which is an acceptable tradeoff in DSM regime. The actual width of the pulse is not critical as long as a minimum duty cycle is maintained across operation [21], [31]. Smaller pulse widths cause less static leakage power. The output voltage $V_{C,}$ when falling from V_{DD} to 0, is given by [28],

$$V_C(t) = V_{DD}.e^{\frac{-t}{(R_d+R_w)C_L}} \tag{1.1}$$

The corresponding capacitor discharge current flowing through interconnect and pull-down resistors is given by the exponential expression,

$$i_C(t) = \frac{V_{DD}}{(R_d+R_w)}\, e^{\frac{-t}{(R_d+R_w)C_L}} \quad (1.2)$$

If the clock period T_{CLK} is sufficiently large to accommodate the transit times, the output capacitor voltage V_C swings rail to rail (0 to V_{DD}). Energy in a cycle can be derived as $C_L V_{DD}^2$ by integrating the instantaneous power ($V_C(t) \times i_C(t)$) over one period T_{CLK} [27]. Then E_{VDD}, the energy drawn from supply per cycle, is $C_L V_{DD}^2$. Similarly, E_C the energy stored in the capacitor can be derived as $C_L V_{DD}^2/2$ [27]. E_C is also the energy dissipated in pull down resistor. For large values of interconnect R_w, the output may not swing rail to rail within the T_{CLK}. In that case, the actual logic high V_{OH} and logic low V_{OL} values can be used, giving a more generalized equation [27] for average power for a clock frequency f_{CLK} (=1/T_{CLK}) as,

$$P_{avg} = V_{DD}\,(V_{OH} - V_{OL})\, C_L f_{CLK}. \quad (1.3)$$

For rail-to-rail operation, the equation for NR operation, valid for all frequencies, is more commonly written as

$$P_{NR} = C_L V_{DD}^2\, f_{CLK}. \quad (1.4)$$

Equation (1.4) is used as a base-line for comparison with other driver schemes. The output is a square wave and does not need special amplifiers to drive flip flops or logic, but may use local clock buffers shown in Figure 1.2. NR supports dynamic voltage and frequency scaling (DVFS) below the maximum operating frequency that the process technology is capable of.

Using (1.4) at 1GHz clock rate, to achieve even a 1V swing on a 1nF capacitor, it takes at least 1W of power [26]. An *LC* resonant global CDN from IBM driving a large load (~6nF) at 4 GHz is integrated in the processor described in [14]. Full functionality over a 20% range in clock frequencies was demonstrated, while saving 6–8 Watts of power that would have been wasted as heat. A similar resonant

grid solution from AMD that saves 25% of the clock distribution power of another high performance processor was reported in [11].

For load capacitor C_L total power dissipation is frequency f times $C_L V_{DD}^2$ [6]. In these resonance schemes, for a given choice of inductor value L, the operating clock range is restricted around the resonance frequency $f = 1/2\pi\sqrt{LC_L}$. The solution is thus tied to one operating clock frequency. It does not maintain the power savings across dynamic voltage and frequency scaling (DVFS).

1.5 Organization of the thesis

The thesis is organized as follows. In Chapter 2, prior-art low power design techniques through energy reuse are formulated, for base line comparisons. Series resonance is examined in Chapter 3, as opposed to more commonly used parallel resonance. The simpler pulsed series resonance (PSR) is detailed first. Simulation results in 45nm CMOS process for clocking operation are shown. Chapter 3 introduces GSR, derived from PSR, as a general purpose solution that can be configured to all other solutions. Simulations validating the design on a 22nm process technology are shown.

In Chapter 4, the support circuitry needed for top down implementation of the clocking schemes using PSR are reviewed. The power losses from the support circuitry and receiving processor units are factored to understand the true overall savings. Previous energy recovery flip-flops are reviewed and true single phase clocking (TSPC) is selected. Circuitry for adaptive pulse generation on both edges of the incoming clock is described. Design of critical circuits needed for the GSR scheme is shown.

Chapter 5 derives timing performance of all drivers. This thesis does a comparative tradeoff analysis of series, parallel and non-resonant topologies for the first time. The implementation details of resonant circuits in deep submicron nodes (DSM) can have implications on area and timing performance.

Chapter 6 shows how the GSR principle can also be extended to data processing circuits using domino-style dynamic logic family with pre-charge mechanisms. Simulation results in 90nm illustrate the power savings achieved by these specialized circuits.

Chapter 7 estimates the active and metal area required by various solutions and other costs of fabrication. No additional area is needed by PSR for dual edge data capture. Complete layout and parasitics are estimated for a 45nm process as an example. Chapter 8 looks at the Power, Performance and Area (PPA) together. The tradeoffs between these for various resonant clocking schemes are discussed. Theoretical performance and power relations of various resonant and non-resonant topologies that can be configured from GSR are tabulated

Chapter 9 shows system integration of PSR clock generation driving 1024 flip-flops through an H-tree. High performance processor benchmark from ISPD2010 clock synthesis contest, drawn from IBM and Intel, in 45nm [32] is used as a test case to demonstrate power reductions. A complete clocking solution with PSR, to minimize power of regional clocks for leaf cells in high performance multi-GHz designs is shown. This novel resonant driver generates pulses at both edges of the square clock for operation in the dual edge mode. Details of simulation results in 45nm CMOS process for clocking and flip-flops are compared. Skew comparison between various schemes shows the advantages of GSR in the performance/price metric. Results for the 22nm node are compared across various schemes.

Chapter 10 discusses a new design flow to incorporate GSR as part of clock tree synthesis to save power and minimize inductance while meeting the timing closure goals. Chapter 11 concludes the thesis with guidelines for extension of this work into the future.

The appendix includes the MATLAB codes for verifying the mathematical derivations used in the chapters. The transistor level schematics of all the circuits used for simulations and test benches are included in the appendices as well.

2 LOW POWER DESIGN THROUGH ENERGY REUSE

One way to reuse the energy $\boldsymbol{C_L V_{DD}^2/2}$ stored on the capacitor, that is wasted away as heat during discharge, is to store it on another storage component and recover it. However, the charging process itself takes $\boldsymbol{C_L V_{DD}^2/2}$ energy, as seen Chapter 1, so that a better means of transfer must be used. An alternative is the so called adiabatic charging using time varying supply voltage. Another method is by using an inductor to transfer the charge on to a capacitor and recover it. Both are explored here.

2.1 Adiabatic Circuits with energy recovery

The set of circuit design techniques targeted at the implementation of computations with minimal (asymptotically zero) power consumption during charge transfer is generally known as adiabatic switching or adiabatic charging. The use of the word adiabatic is suggestive of the thermodynamic principle of state change with no loss of gain or heat. The principle of adiabatic switching can be best explained by contrasting it with conventional dissipative switching.

Figure 2.1(a) shows how energy is dissipated during a switching transition in static CMOS circuits by conventional charging. The transition of a circuit node from LOW to HIGH can be modeled as charging an *RC* tree through a switch, where *C* is the capacitance of the node and *R* is the resistance of the switch and interconnect. When the switch is closed, a high voltage (V_{DD}) is applied across *R* and current starts flowing suddenly through *R*. After a short period of time, *C* is charged to a constant supply voltage V_{DD}. The energy taken from the power supply is CV_{DD}^2, but only half of that , $CV_{DD}^2/2$, is stored in *C*. The other half is dissipated in *R*.

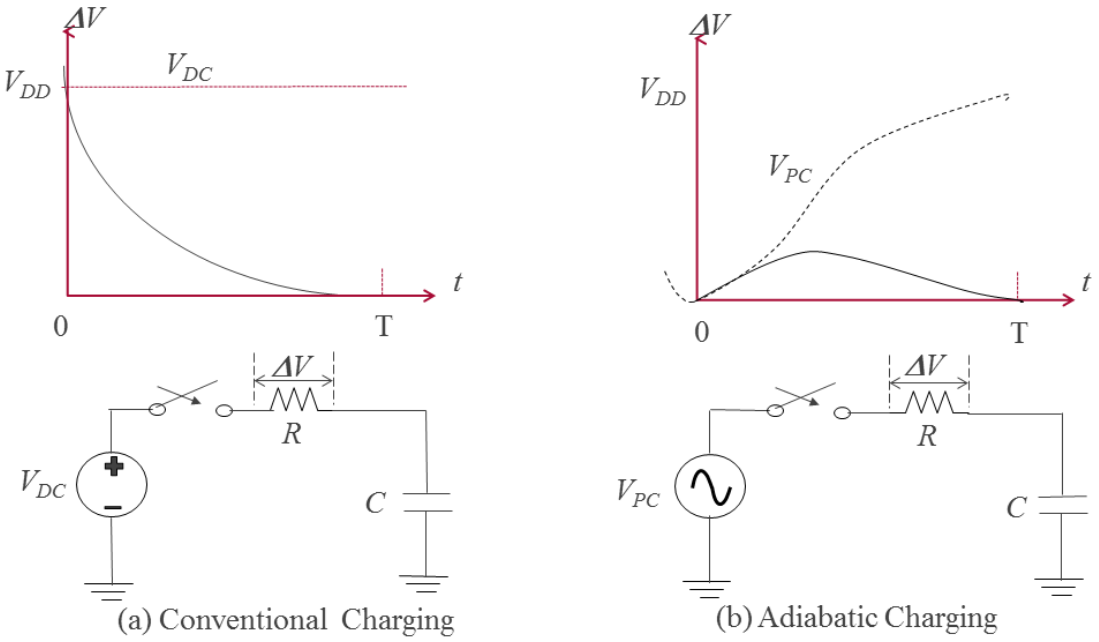


Figure 2.1 Losses in Conventional vs. Adiabatic charging.

Now, consider the circuit and current waveform for adiabatic charging shown in Figure 2.1(b). Notice that, in contrast to conventional charging, the transition has been slowed down by using **a time varying voltage source** (V_{PC}) instead of a fixed supply. By spreading out the charge transfer more evenly over the entire time available, peak current is greatly reduced. The overall energy dissipated E_R in the transition has been shown to have a proportional relationship [9],

$$E_R \propto (RC/T_S)CV_{DD}^2. \qquad (2.1)$$

where R is the effective resistance of the driver device, C is the capacitance to be switched, T_S is the time over which the switching occurs, and V_{DD} is the voltage to be switched across. The constant of proportionality is related to the exact shape of the time-varying voltage source waveform and can be calculated by direct integration.

Ideally, by increasing the time T_S over which computation is performed, it should be possible to create a circuit which computes with vanishingly low energy dissipation as the time allowed for that computation extends indefinitely. Known in

the field as "asymptotically zero energy consumption," practical circuit implementations of these logic elements have been demonstrated [9]. These circuits achieve low, but nonzero, dissipation for computations performed over fixed amounts of time.

Because some of the energy in these circuits (in the form of charge stored on capacitances) was being recovered instead of dissipated, the terms charge recovery or energy recycling began to be used to describe these circuits. Broadly speaking, the term charge recovery is nowadays being used to describe systems that reclaim some of the $C_L V_{DD}^2/2$ energy that is stored in their capacitors during a computation and reused it on subsequent computations.

It should be observed that whenever current experiences a voltage drop ΔV, energy is dissipated at the rate of $i \times \Delta V$ (instantaneous dissipative power), where i is the current. Such energy dissipation can be greatly minimized by deploying adiabatic switching described, where the supply swings gradually from 0 to V_{DD}. There is little voltage drop across the channel of PMOS/NMOS transistor, and hence only a small amount of energy is dissipated. Using simple model of (2.1) to estimate the power dissipation [10], with RC < 1ns for a moderate fan-out, and switch sampling time of $T_S \approx$ l/ f_{CLK} and with an operating clock frequency $f_{CLK} \approx$ 10 MHz, E_R is reduced to a very small value of nearly 1/50th of conventional switching. At higher frequencies of course the savings are less.

Thus, adiabatic charge recovery techniques are very useful in the lower frequency range, like in battery powered systems that need to minimize energy drain. But for clock operation in the GHz range, where the severe heat dissipation occurs in modern processors, other techniques are needed.

2.2 *LC* Resonance Energy Reuse

In this work, the conventional *LC* resonant solutions are termed as CPR solutions since the resonating inductor and capacitor are connected to each other continuously in parallel. A pulsed mode resonant driver is used for driving pulsed flip-flops that can save area and energy.

Figure 2.2 shows the topological comparisons between a non-resonant driver (NR), CPR driver and the new pulsed series resonance driver (PSR). The resonant clocking technique based on Fig 2.2(b) is currently the most commercially viable as it requires minimum change from conventional clock design [14].

The global clock tree can modified to enable resonant (sinusoidal) clocking with an additional metal layer added on top of the conventional tree to attach the inductors and decoupling capacitors [14].

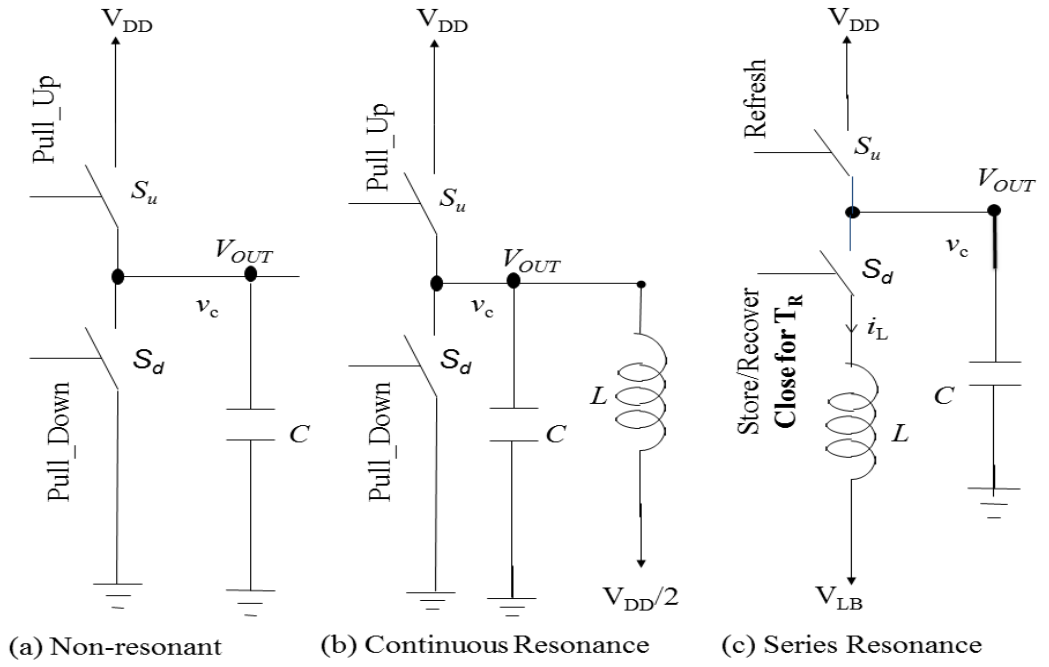


Figure 2.2 Clock Driver Topologies.

2.2.1 Continuous Parallel Resonance Driver (CPR) with Bias Supply

Another way to minimize the capacitor energy discarded is through *LC* resonance. An inductor placed in parallel with the load capacitor minimizes the effective capacitance load at resonance frequency of the *LC* tank formed and can thus reduce the switching energy [22]. Figure 2.3(a) shows a simplified continuous parallel resonant driver (*CPR*) using an extra $V_{DD}/2$ power supply for the inductor. Here the inductor is always connected to load capacitance and the output is a sinusoidal waveform with peak-to-peak reaching twice the bias supply $V_{DD}/2$. In Figure 2.3(b) when the switch S_d is open, it reduces to a parallel *RLC* tank with *Q* given by inductor Q_L at resonance frequency. No PMOS pull up is needed as *LC* tank in resonance will swing twice the inductor voltage $V_{DD}/2$ to give output high of V_{DD}. This scheme has been shown capable of driving the entire clock network of a low power ARM processor [19].

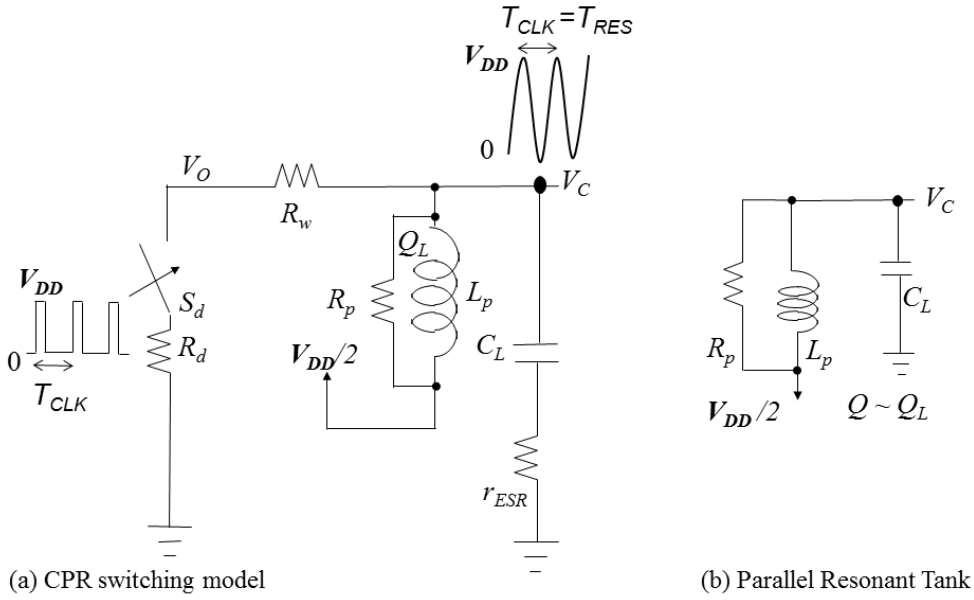


Figure 2.3 Clock Driver Topology for Continuous Parallel Resonance (CPR).

As seen in Figure 2.3(b), a parallel *RLC* network is formed when the grounding switch is opened. R_P at any frequency *f*, comes from the finite quality factor Q_L of

inductor ($R_P = 2\pi f L_p Q_L$). The combined quality factor Q of the tank is determined by the parasitic resistance of the inductor and the equivalent series resistance ESR (r_{ESR}) of the capacitance C_L, if significant [[20], [22]. The ESR is ignored here with respect to R_P, allowing the overall tank Q to be approximated as Q_L.

The general solution for the parallel *RLC* network is obtained from circuit analysis. From Kirchhoff's Current Law (KCL) at node V_C,

$$\frac{V_C(t) - V_{DD}/2}{R_p} + \int \frac{V_C(t) - V_{DD}/2}{L_p} dt + C_L \frac{dV_C(t)}{dt} = 0. \tag{2.2}$$

This leads to second order differential equation for capacitor voltage V_C(t) as,

$$L_p C_L \frac{d^2 V_C}{dt^2} + \frac{L_p}{R_p} \frac{dV_C}{dt} + V_C = \frac{V_{DD}}{2}. \tag{2.3}$$

The initial conditions assumed is $V_C(0) = 0$ with the corresponding initial current in the capacitor $C_L\, dV_C/dt = V_{DD}/2R_p$. Solving using these initial values for complex conjugate roots gives the capacitor voltage V_C as,

$$V_C(t) = \frac{V_{DD}}{2} - \frac{V_{DD}}{2} e^{-\frac{t}{2R_p C_L}} [\cos(2\pi f_R t) - \frac{1}{2Q} \sin(2\pi f_R t)] \tag{2.4}$$

where the damped oscillation frequency $f_R = \frac{1}{2\pi} \sqrt{\frac{1}{L_p C_L} (1 - \frac{1}{4Q^2})}$ and tank $Q = R_p/\sqrt{L_p/C_L}$. This is also called the underdamped case of (2.3) with complex conjugate roots, when $L_p < 4R_p^2 C_L$. As it can be easily seen, a $Q > 0.5$ actually guarantees this condition of underdamped oscillation. At higher values of Q ($> \pi$), f_R is the well-known simpler expression of $f_R = f_{RES} = 1/2\pi\sqrt{L_p C_L}$. At resonance, Q is also given by

$2\pi f_{RES}R_pC_L$. Ignoring the ESR of capacitor, tank Q can be approximated as the inductor component $Q_L = R_P / 2\pi fL_p$.

At high enough frequencies ($f_R >> 1/R_pC_L$), the capacitor voltage from (2.4) in a cycle $0<t<T_{RES}$ can be simplified as [22],

$$V_C(t) \cong \frac{V_{DD}}{2} - \frac{V_{DD}}{2}\cos(2\pi f_{RES}t). \quad (2.5)$$

Since the average DC value is $V_{DD}/2$ on both sides of the resistor R_p, the effective DC power is zero. Thus the CPR power is calculated using power consumed in R_p by the sinusoidal component in (2.5). The average power over one clock period T_{RES} can be obtained from (2.5) as $0.5V_{DD}^2/4R_p$. With $R_P = 2\pi fL_pQ_L = Q/2\pi f_{RES}C_L$ at resonance, CPR power dissipation at resonance can be expressed as [22],

$$P_{CPR} = \frac{\pi}{4Q} C_L V_{DD}^2 f_{RES}. \quad (2.6)$$

Even for a low Q value of π, CPR power is only a quarter of NR power from (2.6). Note that (2.6) as derived is valid only at resonance frequency of operation when $T_{CLK} = T_{RES} = 2\pi\sqrt{L_pC_L}$. For DVFS applications, it is necessary to know how far the operation can be stretched. At clock frequencies above resonance ($T_{CLK} < T_{RES}$), only a portion of the sinusoid in (2.5) is captured. At frequencies below resonance ($T_{CLK} > T_{RES}$) more than one cycle of this sinusoid is captured. The voltage at end of time period T_{CLK}, before being shorted to ground by switch S_d, can be shown as,

$$V_C(T_{CLK}) = \frac{V_{DD}}{2} - \frac{V_{DD}}{2}\cos\left(2\pi\frac{T_{CLK}}{T_{RES}}\right). \quad (2.7)$$

This evaluates to zero for $T_{CLKv}=T_{RES}$ and thus no extra power is consumed other than (2.6). For other frequencies, where output is still periodic and valid, the extra power, coming from discharging the energy stored in the capacitor at voltage $V_C(T_{CLK})$, is $0.5C_LV_C^2(T_{CLK})\, f_{RES}$.

According to (2.7), for $T_{CLK} < 0.5T_{RES}$, less than half the resonant cycle will be captured, making the amplitude lower than $V_{DD}/2$. Similarly, when $T_{CLK} > 1.25\ T_{RES}$, the waveform will cross the midpoint $V_{DD}/2$ and can cause an additional crossover. The corresponding DVFS frequency range for clock signals can thus be approximated to be from $0.8f_{RES}$ to $2f_{RES}$. The average power for this range can again be obtained by integrating $V_C^2(t)/R_p$ from (2.4) over a clock period T_{CLK}, giving a more general equation for power estimation as,

$$P_{CPR} = \frac{\pi}{4Q} C_L V_{DD}^2 f_{RES} + \frac{1}{Q} C_L V_{DD}^2 f_{CLK} - \frac{1}{Q} C_L V_{DD}^2 f_{CLK} \cos^2\left(\pi \frac{f_{RES}}{f_{CLK}}\right) \quad (2.8)$$

At $f_{CLK} = f_{RES}$, (2.8) is same as (2.6). More power is consumed when $f_{CLK} < f_{RES}$ as well as when $f_{CLK} > f_{RES}$, forming a minima at f_{RES}. This behavior is later verified by simulations and also validated by several silicon realizations [11], [12]. The pulse width must be kept sufficiently wide to completely discharge the node through the switch S_d at the given frequency [33], [34]. This is an additional requirement compared to the input pulse stream of NR in Figure 1.4.

Resonant solutions, with characteristic sine wave signals, were initially applied to lower speed systems. Special flip-flops for ultra-low energy applications were designed to work with these low amplitude signals from global clock grids [21]. These custom cells need to be incorporated into standard cell libraries for synthesis.

2.2.2 Parallel Resonance with decoupling Capacitor

The need to meet a high performance clock skew target necessitates the use of a mesh that connects all low skew sinks as shown in Figure 1.2. The capacitance *C* of this grid can be several nFs. The total power dissipation of this NR driver is again

given by (1.4) as, $P_{NR} = CV_{DD}^2 f_{CLK}$. This can be several 10's of watts to meet the stringent skew requirements in high performance designs at GHz clock speeds.

A different implementation for Fig 2.2(b) CPR driver, with the inductor bias supply replaced by capacitors, is shown in Figure 2.4. The inductor bias voltage source is eliminated with use of large capacitors, but 50% duty cycle inputs are needed and lower power savings are obtained. Within Figure 2.4 the parallel *R, L* and *C* network has a combined tank quality factor $Q=2\pi f_{CLK} R_P C$ at resonance (i.e. $f_{CLK}=1/2\pi\sqrt{LC}$) with $R_P = Q/2\pi f_{CLK} C$. With inductor biased at $0.5V_{DD}$ and with V_{OUT}(t) intially at $V_{DD,}$ the resonant clock output signal can be solved for, similar to (2.5), to give the equation,

$$V_{OUT}(t) = 0.5V_{DD} + 0.5V_{DD} \cos(2\pi f_{RES} t) \quad (2.9)$$

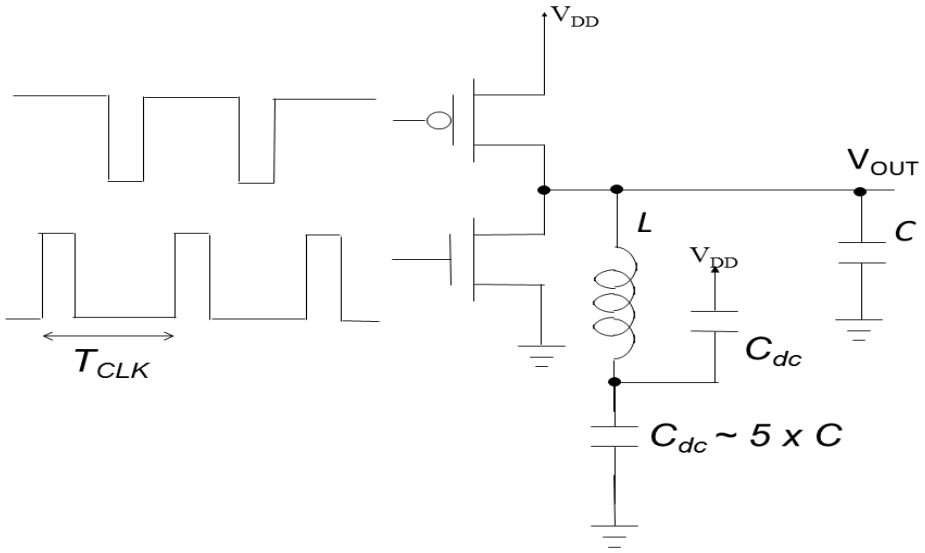


Figure 2.4 Conventional Continuous LC Resonant Clocking Driver (CPR).

Resonant clocks have also been synonymously referred to as sinusoidal clocks [11] due to the waveform from (2.9). The indcutor current can be shown to be,

$$i_L(t) = I_o \sin(2\pi f_{RES} t) \quad (2.10)$$

where the peak current $I_o = 0.5V_{DD}/\sqrt{L/C}$.

Power is dissipated only in the equivalent resistor R_P. The DC component of power is $0.5V_{dd}$ in R_P given as $V_{DD}^2/4\ R_P$. The AC power due to a sinusoidal component of $0.5V_{DD}$ amplitude is $0.5(0.5V_{DD})^2/R_P$. Thus the total DC and AC power consumption is $1.5V_{DD}^2/4R_P$. Substituting for $R_P = Q/2\pi f_{RES}C$, the power dissipation with decoupling capacitance can be expressed as,

$$P_{CPR\text{-}C} = 1.5 \times 2\pi f_{RES}\, CV_{DD}^2/4Q = (3\pi/4Q)\, CV_{DD}^2 f_{RES} \qquad (2.11)$$

This is assuming resonance at $f_{CLK} = f_{RES}$, resulting in pure sinusoidal outputs that would take minimum power. Q is the combined quality factor of inductor and load capacitor [24]. It accounts for the *ESR* of the capacitor and the DC resistance (*DCR*) of the inductor. Even for realizable low Q values like π, CPR power will only be ¾ of NR power for global CDNs. CPR based global CDNs have been reported to yield 25% or more power reductions [11].

Additional chip area occupied by the inductor may not be acceptable, especially for low load capacitance values of 1pF or less. As the resonance frequency is set by $f_{CLK} = 1/2\pi\sqrt{LC}$, different inductor values are needed to operate at different frequencies. This makes it incompatible to DVFS unless the inductors are changed on the fly [1]. Moreover, at frequencies 2× lower than resonance, waveforms get warped and the skew suffers as well [24]. While the CPR can be disconnected at these frequencies, the power savings will not be available [11]. As Figure 2.4 shows, large decoupling capacitors are needed for CPR schemes to hold $V_{DD}/2$ center bias. This takes couple of cycles of clock before settling to the final value. Thus clock gating, to shut down switching power dynamically, is not possible with this scheme as th.. use e driver is expected to be functional without any cycle slips at turn on.

LC resonant circuit operation can reduce the buffer sizes as well. This reduces the total load capacitance and lowers the power further. Hence, in spite of several issues discussed above, CPR CDNs are attractive at global clock level. Usually, local gates and flip-flops in a sector are buffered by local clock buffers (LCB). The clock signal feeding the registers, as shown in the bottom of Figure 1.2, is a square (wave) clock. Inserting inverters in the clock path eliminates the energy recovery property. If the bulk of the CDN capacitance is in its leaves, then the largest power advantage will come by extending the resonance down to the flip-flops. In [21],[23], [24] the clock buffers are removed to allow the clock energy to resonate between the inductor and the local clock capacitance.

3 SERIES RESONANCE FOR WIDE FREQUENCY CLOCKING

This chapter arrives at the new configurable Generalized Series Resonance (GSR) and shows how various clock driver schemes to drive large capacitive loads can be derived from it. The theoretical tradeoffs between various resonance solutions are analyzed so that the optimum configuration may be selected for the given application.

3.1 Pulsed Series Resonance (PSR)

Another way to use an inductor to save energy stored on a large load capacitance is shown in the resonant topology of Figure 3.1(a), where the inductor is periodically connected to load capacitance with controlled input pulse width T_{PW}. Output has a pulse of width T_{RES} driving a higher capacitive load at resonance. For ideal inductor (Q_L >> 10), both input and output are from 0 to V_{DD}. Figure 3.1 (b) shows series RLC model for analysis with bottom switch S_r closed and top switch S_u open during time 0 to T_{PW}. The implementation was presented in ISCAS2014 [23] and the theoretical analysis with performance trade-off equations is detailed here. Compared to CPR in Figure 2.3, the inductor is moved from the output to bottom of switch S_r. Controlled by the pulses of *PLS_CLK* signal, S_r closes when output needs to go low. The series inductor allows the energy stored on the load capacitor to be transferred to the $V_{DD}/2$ node and then recovered back immediately to make the output go high. This creates a pulse of resonance period T_{RES}. Energy can be recycled with the series LC resonant tank ($f_{RES}=1/2\pi\sqrt{L_S C_L}$) formed in Figure 3.1(b) when S_r is closed [23], [24]. Thus, the pull-up switch does not need to charge the output to V_{DD} all the way from 0V. Such a pulsed series resonance (PSR) topology can also use bond wire inductors or off-chip inductors [24].

The input stream *PLS_CLK* is required to have certain width (T_{PW}), as shown in Figure 3.2(a), to generate a resonant pulse stream at the output [24]. Figure 3.2(b) shows the output timing waveforms for the PSR circuit. The energy recovery process is done through the inductor current in resonant mode.

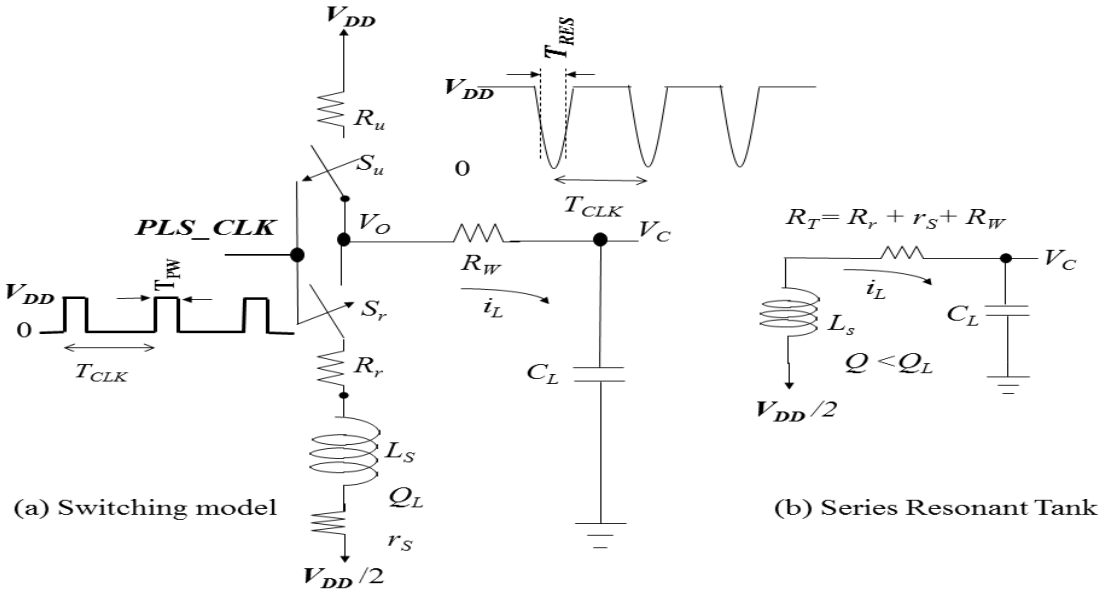


Figure 3.1 Pulsed Series Resonance (PSR) (a) Switching Circuit (b) Linear Model

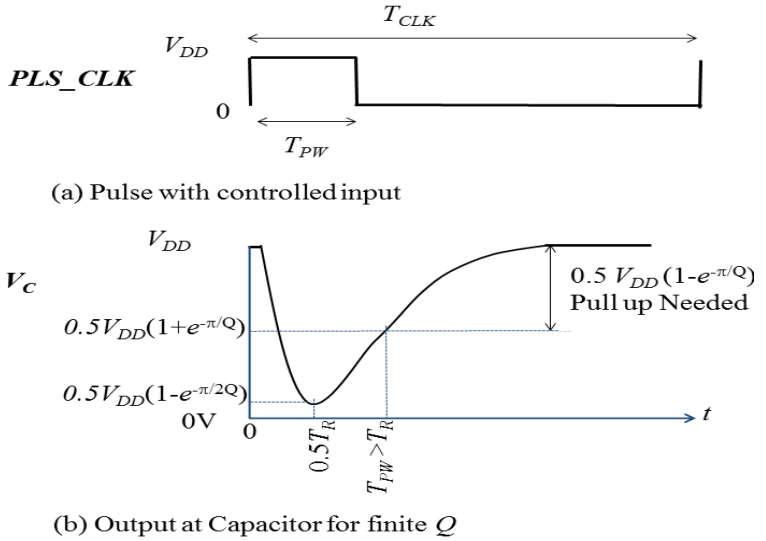


Figure 3.2 PSR Operation with losses. (a) Input pulse (b) Output pulse.

When input signal *PLS_CLK* is high, the resonant tank is formed and when low, the driver is in non-resonant mode. Unlike in CPR, there is an extra requirement on keeping the incoming pulse width T_{PW} exactly related to T_{RES}, across all operating frequencies, for a given C_L and L_S. The resonance time is $T_{RES} = 2\pi\sqrt{L_S C_L} < T_{CLK}$. This inequality requirement, rather than equality in CPR, between C_L, L_S and T_{CLK} values provides an extra degree of freedom. Several advantages result from this as described later in Chapter 8.1.

When operating with narrow output pulses, T_{RES} is always less than the period T_{CLK} across DVFS. The *PLS_CLK* signal with required T_{PW} can be derived from the regular clock using circuitry shown in Chapter 4. Analysis of Figure 3.1(b) is first done for a step input from the closing of the S_r (NMOS) switch.

In Figure 3.1(b), the total resistance is the series combination $R_T = (R_r+R_W+r_S)$. Here $r_S = 2\pi f L_S/Q_L$ is from the finite Q_L of inductor at frequency *f*, and can include the output impedance of $V_{DD}/2$ supply as well [23], [35]. The parasitic equivalent series resistance (*ESR*) of the load capacitance is ignored in this comparative analysis, but can be factored as the component quality factor Q_C. Thus, the overall tank $Q = 2\pi f L_S/R_T$ is degraded, as R_T is larger than r_S.

The loop in Figure 3.1 (b) yields,

$$R_T i_L(t) + \int \frac{i_L(t)}{C_L} dt + L_S \frac{dVi_L(t)}{dt} = \frac{V_{DD}}{2} \tag{3.1}$$

This leads to second order differential equation for inductor current i_L (t) with initial condition i_L (t) = 0 and $\frac{di_L}{dt} = 0$ as,

$$\frac{d^2 i_L}{dt^2} + \frac{R_T}{L_S}\frac{di_L}{dt} + \frac{i_L}{L_S C_L} = 0 \tag{3.2}$$

For underdamped case having complex conjugate roots, the inductance needs to have minimum value given by condition $L_S > R_T^2 C_L/4$ [7], [24]. The solution for (3.2) would then give the inductor current as,

$$i_L(t) = \frac{V_{DD}}{2\sqrt{L_S/C_L}\sqrt{1-\frac{1}{4Q^2}}} e^{-tR_T/2L_S} \sin(2\pi f_R t) \tag{3.3}$$

where the damped oscillation frequency f_R is given by,

$$f_R = \frac{1}{2\pi}\sqrt{\frac{1}{L_S C_L} - \frac{R_T^2}{4L_S^2}} = f_{RES}\sqrt{1-\frac{1}{4Q^2}} \tag{3.4}$$

and the tank Q by $\sqrt{L_S/C_L}/R_T$. The currents peaks are between $\pm V_{DD}/2\sqrt{L_S/C_L}$.

Assuming $1/f_R \leq T_{PW} < T_{CLK}$, the capacitor output voltage can be derived by integrating the current in the capacitor to give,

$$V_C(t) = \frac{V_{DD}}{2} + \frac{V_{DD}}{2} e^{-tR_T/2L_S}[\cos(2\pi f_R t) - \frac{1}{2Q}\sin(2\pi f_R t)]. \tag{3.5}$$

For large tank Q values the two frequencies f_R and f_{RES} can be taken as equal. At resonance, the *RLC* tank $Q=2\pi f_R R_T C_L$, is also large when underdamped case is met. The last term in (3.5) can also be neglected for large Q values.

In Figure 3.2(a) an input pulse stream required at clock frequency with controlled pulse width T_{PW}. Figure 3.2(b) shows output pulse with non-deal inductor (Q_L < 10) when cycling though one clock period. Input pulse width T_{PW} must be larger than damped oscillation cycle T_R. Voltage V_C on the capacitor (Q_C > 30) does not swing rail-to-rail. Extra power is needed to restore V_C to V_{DD} rail. If the width of input pulses (T_{PW}) is sufficient to allow the inductor current waveform to go through a complete resonance cycle $T_R = 1/f_R$, all the possible energy can be recovered. The output voltage rises to high by itself till a certain voltage recovery point, without drawing current from V_{DD} power supply. The charging and discharging waveforms are actually adiabatic in nature, thus minimizing transfer losses.

The underdamped capacitor output will ring with minimum value at $t = T_R/2$. The first maximum is at $t = T_R$, giving rise to the waveform in Figure 3.2(b). Substituting from the *RLC* series resonance Q expression $R_T/L_S = 2\pi f/Q$, the first maximum value at $t = T_R$ from (3.5) can be expressed as,

$$V_{OH} = 0.5V_{DD}(1+e^{-\pi/Q}). \tag{3.6}$$

To reach 90% of V_{DD}, as normally required, a $Q \geq 14$ is needed. As this is generally too high to realize on chip, the output is pulled up to rail using the S_u (PMOS) switch, forcing the final V_{OH} to V_{DD}.

Similarly, the minimum voltage logic low V_{OL} can be calculated from (3.5) at $T_R/2$ as,

$$V_{OL} = 0.5V_{DD}(1-e^{-\pi/2Q}). \tag{3.7}$$

To reach the standard 10% of V_{DD}, a $Q \geq 7$ is needed. This is less difficult to achieve than V_{OH} requirement. Lower V_{OL} can also be obtained by using an inductor bias lower than $V_{DD}/2$. This will also change (3.1) and (3.5) giving a lower V_{OH} than (3.6), but is taken care of by pull up switch S_u. As shown in Figure 3.2(b), the highest voltage recovery point from freewheeling resonance oscillation is less than V_{DD}. Thus power needed to pull it from this to full V_{DD} swing on C_L at frequency f_{CLK} can be obtained similar to (1.3) as,

$$\begin{aligned} P_{PSR} &= V_{DD}\,(V_{DD} - 0.5\,V_{DD}\,(1+e^{-\pi/Q}))\,C_L f_{CLK} \\ &= 0.5\,(1-e^{-\pi/Q})\,V_{DD}^2\,C_L f_{CLK}\,. \end{aligned} \tag{3.8}$$

This is valid for all frequencies where $f_{CLK} < f_R$ and not just at resonance like CPR. At $Q = \pi$, PSR takes about 1/3 power of NR. While the power savings are seemingly lower than CPR the advantage is that, they are realized at all DVFS frequencies.

3.2 Generalized Series Resonance

Figure 3.3 show a series resonance scheme generalized from PSR [23], [26] and termed here as GSR. Figure 3.3(a) shows Generalized Series Resonance (GSR) with pull up and pull down switches for rail-to-rail operation. Figure 3.3(b) shows an equivalent series resonant circuit model for GSR with S_r closed, *Su* open and S_d open. The output of PSR is a narrow pulse stream rather than near 50% duty cycle of standard clocks.

Figure 3.4 shows the required timing diagram for generating rail-to-rail (0 to V_{DD}) clock output pulses crucial for controlling the switching operation in GSR. The equal pulse widths of V_{SR} generated from rising and falling edges of the clock input can be used to logically derive the switch control signals $\overline{V_{UP}}$ and V_{DN} to generate ideal 50% duty cycle output clock at V_C.

All voltage signals swing 0-V_{DD}. The i_L current peaks are $\simeq \pm V_{DD}/2\sqrt{L_S/C_L}$. With switch control timing shown in Figure 3.4, outputs with duty cycle close to 50% are obtained in GSR. As the values of Q are very low (< 4) on-chip, the V_{OH} of PSR is be improved from (3.6) by using a separate pull up switch S_u in Figure 3.1.

Additionally, the V_{OL} can be improved from (3.7) with a pull down switch S_d. GSR has the extra pull down switch S_d to give rail-to-rail operation. This new GSR topology in Figure 3.3 has independent control nodes for switches S_u and S_d, like NR of Figure 1.4. The active high control signal V_{SR} is derived (as shown later in Chapter 4) from both edges of the incoming 50% duty cycle clock.

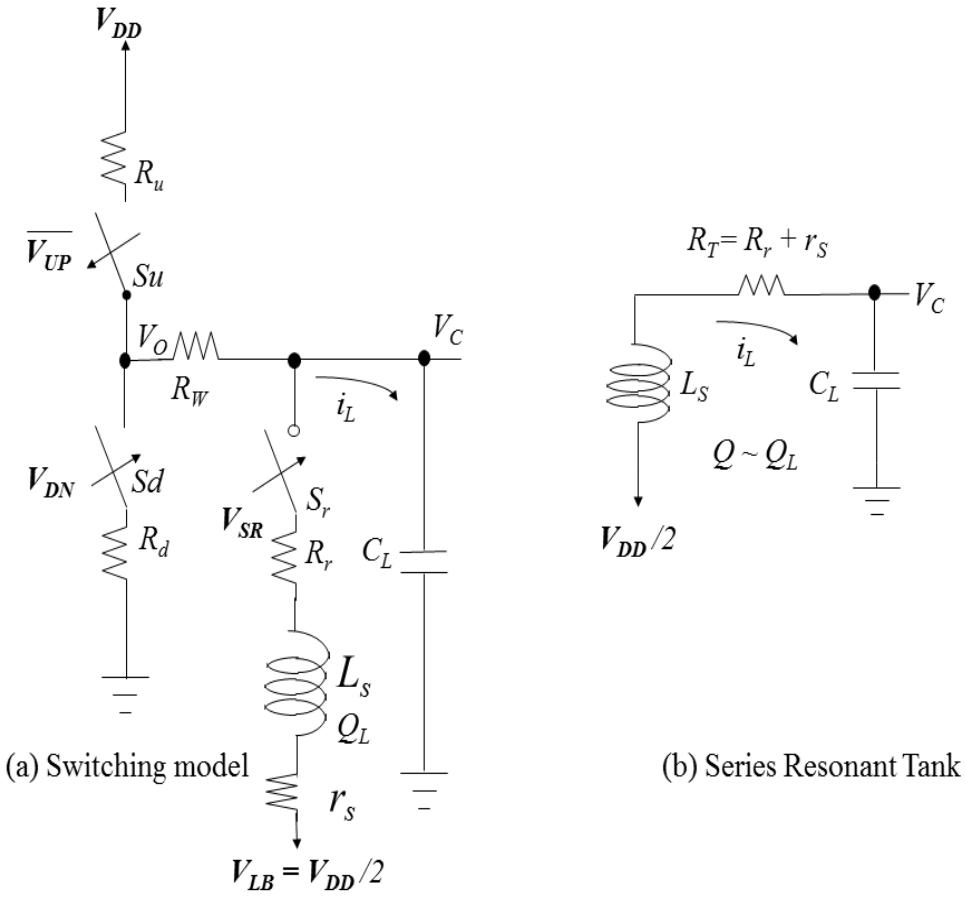


Figure 3.3 GSR (a) Switching circuit (b) Equivalent circuit model

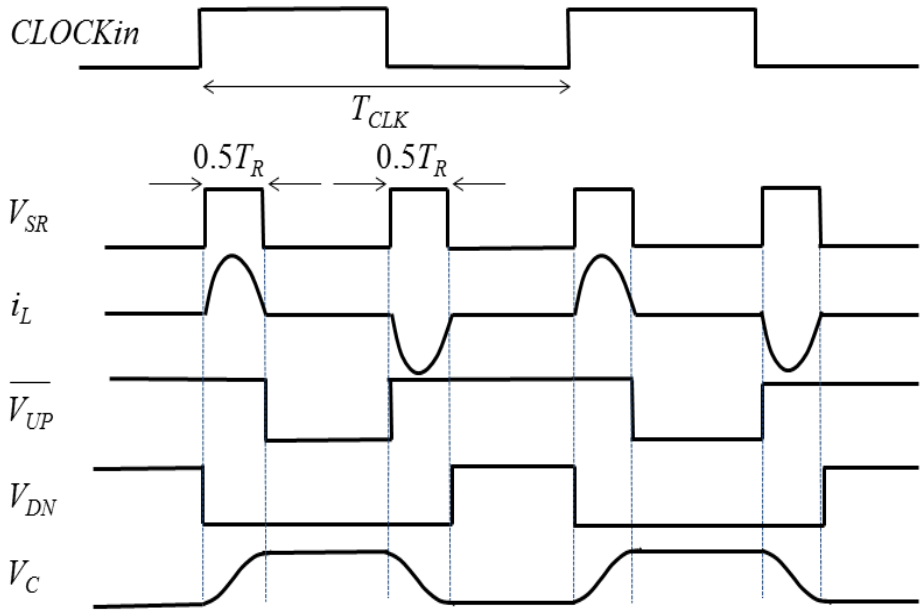


Figure 3.4 Timing diagram for generating rail-to-rail clock output.

The switch S_r in series with inductor is closed twice in a cycle, first to store the discharging energy and later to recover it. The switch S_r control input pulse stream V_{SR} needs to have a specific width ($T_R/2$) for resonance. The active low $\overline{V_{UP}}$, after resonant recovery during V_{SR} pulse, pulls up the output to V_{DD}. The active high V_{DN} signal pulls down the low going output signal all the way to ground, after the V_{SR} pulse. As seen by (3.7), for low Q, the output does not go all way to bottom rail with resonant discharge.

Adiabatic transfer of the energy between the inductor and load capacitor during the resonance periods effectively conserves dynamic energy. Compared to PSR, the inductor in GSR is switched at twice the rate ($2f_{CLK}$) of the incoming clock and for half the duration ($T_{PW} \approx T_R/2$). The governing equations during S_r closure are same as (3.1) and (3.2) derived for PSR, but with different initial conditions. The inductor current is then given by (3.3) and capacitor voltage by (3.5). However, the waveforms last only for half the cycle. The energy recovery process can be seen from the ideal inductor current i_L into the V_C node, where the current during discharge is recovered back for charging.

When V_{SR} pulse closes S_r for half the resonance period, the V_C is discharged to lowest point $0.5V_{DD}(1-e^{-\pi/2Q})$ from (3.7). The switch is ideally opened when the current is zero and charge stored on the $V_{DD}/2$ node. The V_{DN} signal then closes, connecting output to ground and forcing the V_{OL} to 0V rail. When the V_{SR} pulse comes next in charging phase, it will follow (3.5) again with a half cycle time shift starting from 0V. It will not reach the PSR maximum recovery point V_{OH} but will be shifted down by V_{OL}. This will give maximum resonance recovery point rising from ground as,

$$V_{Cmax} = V_{OH} - V_{OL} = 0.5\, V_{DD}\,(1+e^{-\pi/Q}) - 0.5V_{DD}(1-e^{-\pi/2Q})$$
$$= 0.5V_{DD}\,(e^{-\pi/Q} + e^{-\pi/2Q}). \quad (3.9)$$

When the V_{UP} signal is active, it will pull up from the V_{Cmax} value to V_{DD}. From (3.9), it can be seen that the voltage recovery point is lower than in PSR (3.6), requiring more energy to replenish, for the rail-to-rail operation.

The power needed in P_{GSR} to pull V_C from the value in (3.9) to V_{DD} at frequency f_{CLK} can be derived similar to (3.8) as,

$$P_{GSR} = (V_{DD} - V_{Cmax})\, V_{DD}\, C_L f_{CLK}$$
$$= (V_{DD} - 0.5\, V_{DD}e^{-\pi/2Q} - 0.5\, V_{DD}e^{-\pi/Q})\, V_{DD}\, C_L f_{CLK} \quad (3.10)$$
$$= (1 - 0.5\, e^{-\pi/2Q} - 0.5e^{-\pi/Q})\, C_L V_{DD}^2 f_{CLK}.$$

By connecting the inductor branch closer to the load, the series resonance total resistance can be reduced to $R_T = (R_r + r_s)$. This will prevent significant Q degradations, improving the energy savings further. The same assumption is made, as in PSR, $4L_S/R_T > R_T C_L$ for underdamped condition, implying a minimum value of inductance and Q.

The power is less than that taken by NR and, for a Q of π, nearly 50% savings is predicted. GSR savings are valid over DVFS clock frequency range. The tank Q for GSR can be maximized as the inductor is free to be connected closer to C_L.

3.3 GSR with decoupling capacitor (GSR-C)

It is also possible to use GSR with a large decoupling capacitor instead of the extra inductor bias supply, like in CPR, as shown in Figure 3.5. An energy recovery capacitor C_{ER}, is incorporated for electrical energy storage and initializing the logic operation as shown in Figure 3.5(a). Figure 3.5(b) shows an equivalent series resonant circuit model for energy conserving clocking circuit.

The resonant circuit incorporates a high-Q inductor L_S connected in series with capacitors C_{ER} and C_L, along with switching transistors. R_r is the ON resistance in the FET switch, when operating in the linear regime, and r_s is the inductor series resistance in the resonant circuit. The equivalent circuit of series *RLC* resonator with R_r, r_s and L_S connected in series with capacitors C_{ER} and C_L is shown in Figure 3.5(b).

A virtual voltage source is created by adding the energy-recovery storage capacitor C_{ER} in the circuit. This capacitor is precharged to a voltage of $V_{DD}/2$ to begin with. The restoring voltage $V_{DD}/2$ in the storage capacitor C_{ER} is assumed to be stable during the charging and discharging of C_L. This energy conserving resonant circuit is used for generating flat-topped (trapezoidal) clocking waveform with a very low energy loss.

Figure 3.6 shows the timing diagram for generating the flat-topped output pulses by the energy recovery logic circuit. The period of clocked waveform can be determined by the designed values of inductance and capacitances in a circuit. Here, the energy recovery capacitor C_{ER} is used as a reservoir, as energy moves back and forth to load capacitor C_L. Current flows into the load capacitor and a voltage is generated in a series inductor, $L_{S.}$ When the output voltage V_C reaches the same potential ($V_{DD}/2$) as the storage capacitor C_{ER}, the voltage across the inductor begins to collapse and the current is forced to flow in the same direction through the inductor L_S, forcing the V_C to approach V_{DD}.

The output voltage V_C reaches V_{DD} at the point when the current i_L in the series inductor becomes zero. At this time, the switch S_r is turned off, and the switch S_u is turned on. This holds the output voltage at V_{DD} for finite time, giving the flat-top of the output pulse. Energy is wasted through switch S_u to bring the output to the full logic ‘1’, and replenish any energy dissipated in the resonant circuit.

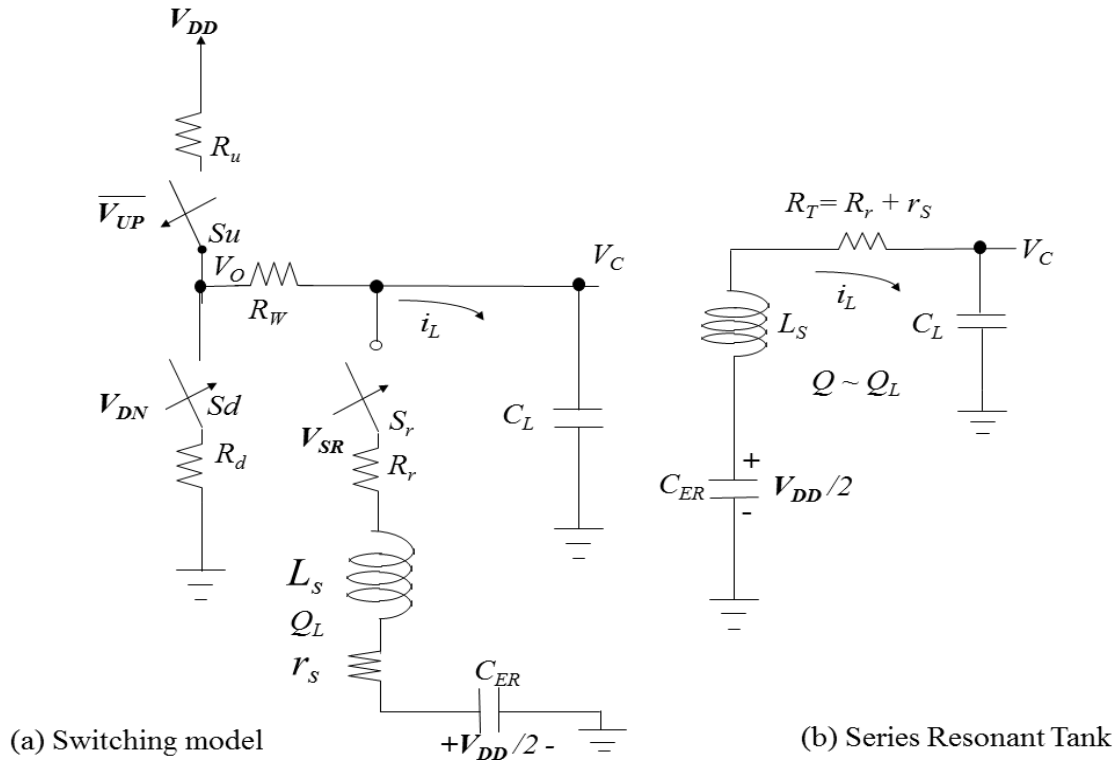


Figure 3.5 GSR-C with energy recovery capacitance C_{ER}.

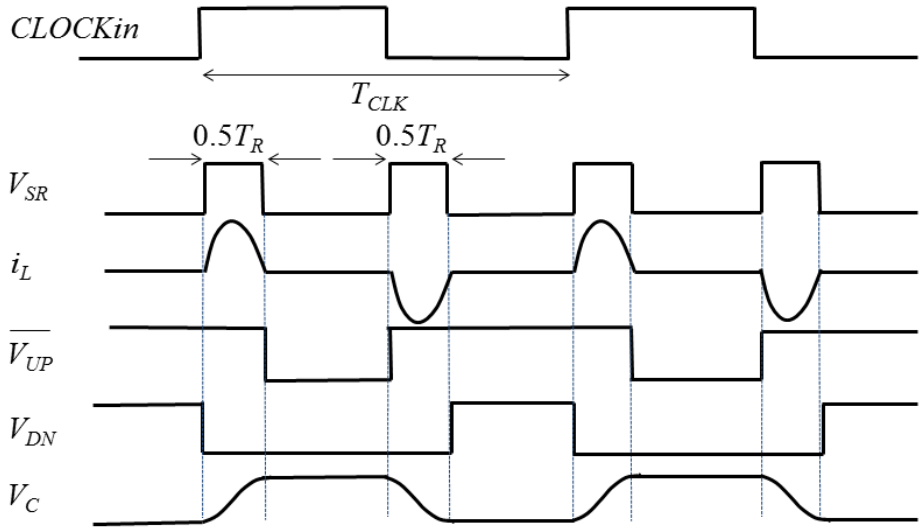


Figure 3.6 Same as Figure 3.4, repeated for convenience.

In the discharge phase, the charge is returned to the energy-recovery biasing capacitor C_{ER} by current flowing through the inductor by turning on the switch S_r. Other switches S_u and S_d are turned off. C_{ER}, L_S and C_L again form a series resonant

circuit for energy transfer from to the inductor by current flowing out of C_L through L_S. This causes a build-up of voltage in L_S in the direction opposite to the charging phase, returning charge to the capacitor C_{ER}. When V_C decreases to $V_{DD}/2$, the voltage of L_S collapses forcing the current in same direction, making V_C reach ground at logic '0'. At this point, the current i_L becomes zero and the switch S_r is turned off, and the switch S_d is turned on to hold the output voltage to ground (i.e., logic '0'). Through this resonant energy transfer mechanism, most of the energy is recovered. The charge is restored to the biasing capacitor C_{ER}, and a stable $V_{DD}/2$ stored voltage is maintained during the circuit operation. For this, the designed value of C_{ER} is kept much larger than C_L. In this way, the resonant driver with controlled switches generates a sequence of output voltage pulses with finite flat-tops.

The control signals for the switches are almost identical to the ones in Figure 3.4 and repeated here for convenience. The loop in Figure 3.5(b) from Kirchhoff's Voltage Law (KVL) yields,

$$R_T i_L(t) + L_S \frac{di_L(t)}{dt} + \int \frac{i_L(t)}{C_{ER}} dt + \int \frac{i_L(t)}{C_L} dt = \frac{V_{DD}}{2} \tag{3.11}$$

This leads to second order differential equation for inductor current i_L (t) with initial conditions i_L (t) = 0 and $V_C(0) = V_{DD}/2$ as,

$$\frac{d^2 i_L}{dt^2} + \frac{R_T}{L_S}\frac{di_L}{dt} + \left(\frac{C_L + C_{ER}}{C_L . C_{ER}}\right)\frac{i_L}{L_S C_L} = 0 \tag{3.12}$$

The combined total capacitance can be designated as $C_T = \frac{C_L + C_{ER}}{C_L . C_{ER}}$. For underdamped case having complex conjugate roots, the inductance needs to have minimum value given by condition $L_S > R_T^2 C_L/4$ [7], [24]. The solution for (3.2) would then give the inductor current as,

$$i_L(t) = \frac{V_{DD}}{2\sqrt{L_S/C_L}\sqrt{1-\frac{1}{4Q^2}}}\sqrt{\left(1+\frac{C_L}{C_{ER}}\right)}e^{-tR_T/2L_S}\sin(2\pi f_R t) \quad (3.13)$$

where the damped oscillation frequency f_R is given by,

$$f_R = \frac{1}{2\pi}\sqrt{\frac{1}{L_S C_T} - \frac{R_T^2}{4L_S^2}} = f_{RES}\sqrt{1-\frac{1}{4Q^2}} \quad (3.14)$$

with $f_{RES} = \frac{1}{2\pi}\sqrt{\frac{1}{L_S C_T}}$, with C_T being the total series capacitince $\frac{C_L+C_{ER}}{C_L.C_{ER}}$.

Intrgrating the current throught the capacitance $C_{L,}$ one can obtain the voltage as,

$$V_C(t) = \frac{V_{DD}}{2}e^{-tR_T/2L_S}\left[\cos(2\pi f_R t) - \frac{1}{\sqrt{4Q^2-1}}\sin(2\pi f_R t)\right]. \quad (3.15)$$

Energy disspiation occurs because of the resistive losses and this can also be obtained by intergrating instantaneous power $i_L(t) \times R_T$ over a cycle to yield the disspiation and power as,

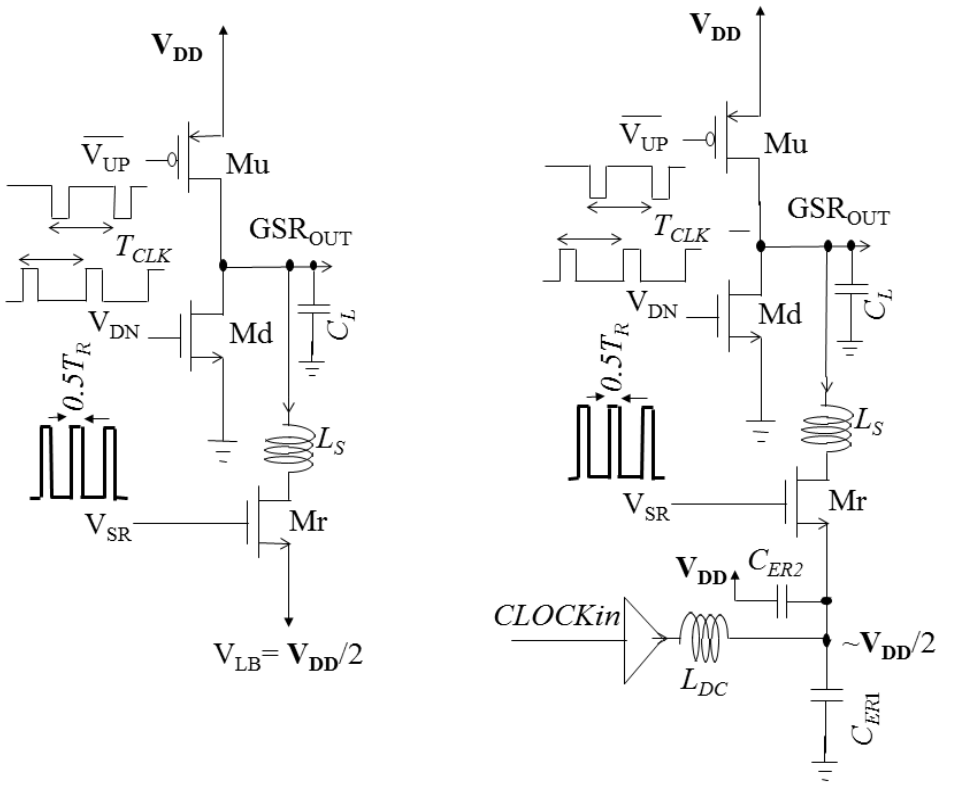
$$\mathrm{P_{GSR\text{-}C}} = \frac{1}{4}C_L V_{DD}^2 f_{CLK}\left(1+\frac{C_L}{C_{ER}}\right)\left(1-e^{-\frac{2\pi}{\sqrt{4Q^2-1}}}\right) \quad (3.16)$$

The power is less than that taken by NR and, for a Q of π, nearly 80% savings is predicted with C_{ER}>10×C_L. There is of course an area penalty for using this scheme. GSR-C savings are valid over DVFS clock frequency range. The tank Q for GSR-C can also be maximized as the inductor is free to be connected closer to C_L. However, the C_{ER} capacitor needs to be placed close to the inductor so that there are routing challenges for this during integration. Also clock gating is not possible with GSR-C

3.4 GSR Transistor level configurations

Figure 3.7 shows transistor level implementation of the GSR driver output stage with the all the incoming control signals. In the case of the scheme (a) separate

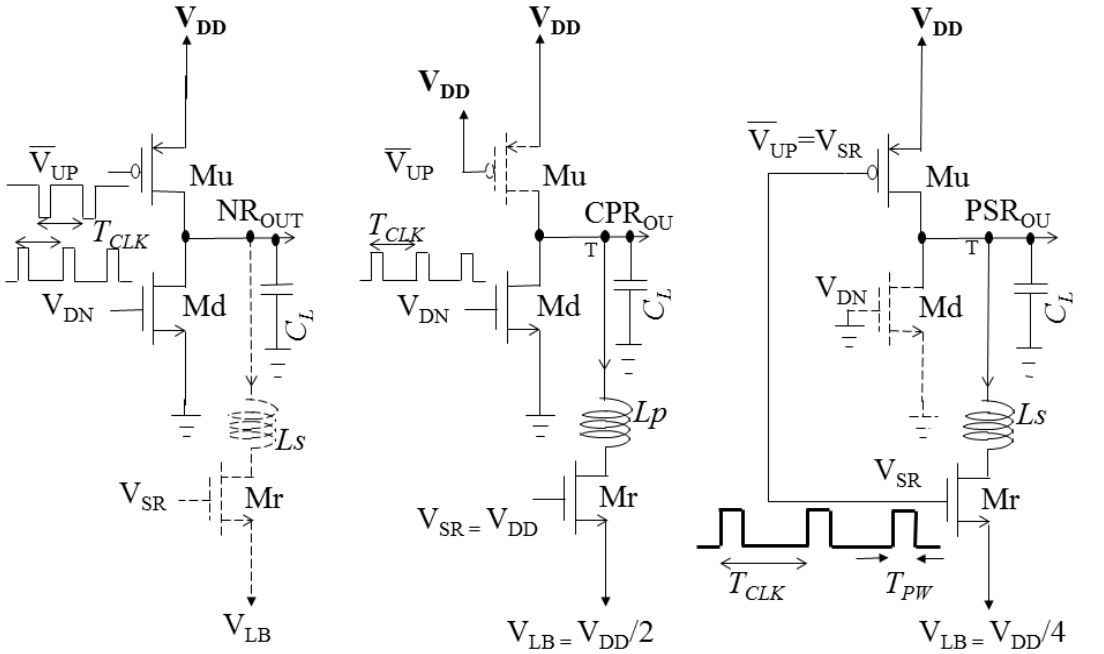
inductor bias supply is used. Figure 3.7(b) uses a large capacitor. The clock input is buffered and filtered to pre-bias the line as needed. The capacitor is charged to mid voltage $V_{DD}/2$ by filtering a buffered version of the input clock signal, that is usually 50% duty cycle. Inductor L_{DC} is kept 10-100 times L_S as practical. Capacitors C_{ER1} and C_{ER2} are taken to be 5 times C_L. The input clock to this generator may be gated as needed to reduce the extra power consumption.



(a) GSR full configuration with Bias Voltage (b) GSR full configuration with Capacitor Bias

Figure 3.7 GSR full configurations.

Figure 3.8 shows three possible reconfigurations of the GSR to give NR, CPR and PSR modes. The NR schemes does not need *Mr* transistor and can thus be turned off. The CPR scheme similarly does not need *Mu* and this can be tied off. PSR does not need *Md* and this can be disabled too. These reconfigurations can also be done dynamically to achieve best system level performance depending on the application.



(a) NR configuration with V_{SR}=0, V_{LB}=0

(b) CPR configuration with V_{SR}=V_{DD}, V_{LB}=V_{DD}/2, V_{UP}=V_{DD}

(c) PSR configuration with V_{LB}=V_{DD}/4, V_{DN}=0, V_{UP}=V_{SR}

Figure 3.8 GSR Reconfigurations.

3.5 Series Resonance Simulation Results

Using the transistor level configurations of Figure 3.8, PSR and GSR configurations are simulated to verify the basic functionality and power savings derived in theory.

3.5.1 PSR Functionality

The resonance time, designated as T_{RES}, is given by $2\pi\sqrt{LC}$. T_{PW} should thus ideally be of T_{RES} duration, basically the period of resonance for large Q. This period (T_{RES}=1/f_{RES}) is set at a third of maximum T_{CLK} or even less. As an example, for a 1pF load at 1GHz clock rate, T_{RES} can be set to 0.2ns using a 1nH inductor resulting in 5GHz resonance frequency. Conventional CR would need 25nH to resonate with a 1pF load. As the inductor is not continuously connected to the output it only needs a

global bias line V_{LB}. Figure 3.9 shows the basic operation of PSR for a 1GHz clock in a 45nm IBM compatible process [36], [37].

There is some ringing in the current that can be observed when the inductor is disconnected and left floating in the non-resonant portion as T_{PW} is larger than T_{RES}. This is actually necessary to conserve energy. The performance must be viewed along with data capture of flip-flops as shown in Chapter 9.

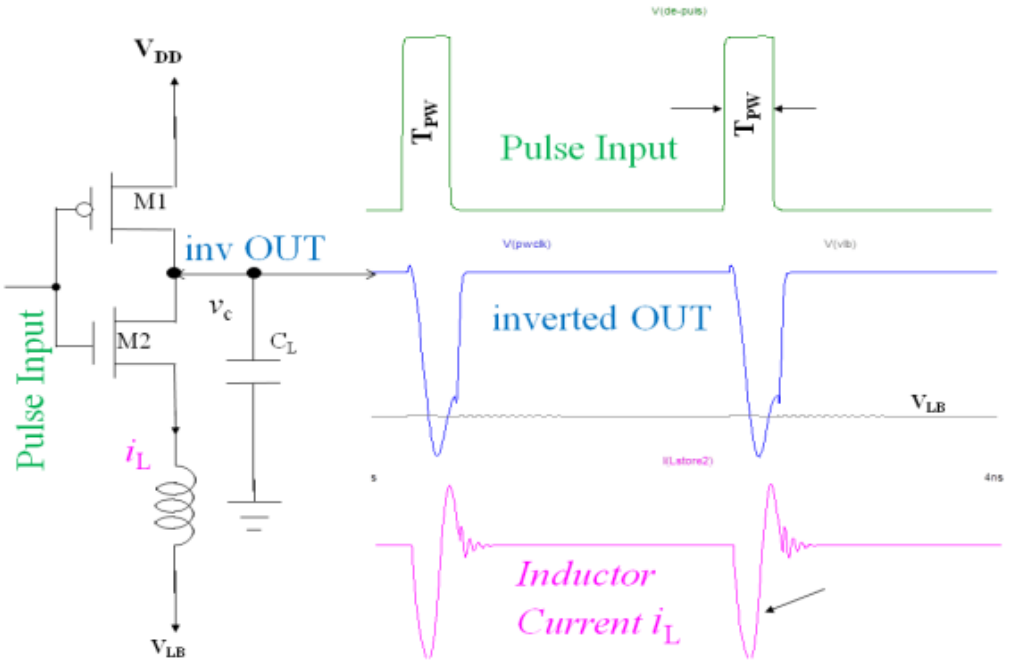


Figure 3.9 PSR Operation Timing Waveforms.

3.5.2 GSR Functionality and Performance

The functionality and robustness of the new GSR and GSR-C drivers is verified by 22nm SPICE simulations [36], [37]. The results plotted in Figure 3.10 show that, the GSR (red) and GSR-C (blue) output V_C are functional to drive standard local buffers generating an output signal for flip flops and other parts of the digital system. The pulse width of V_{SR} varies to track the changes in the *LC* resonance time

that come from variations in load capacitance. The V_{SR} , pull up V_{UP} and pull down V_{DN} signals are shown later in Chapter 4. The bias voltages are shown for the two different schemes. Although GSR-C generates less than $V_{DD}/2$ bias through the filtering, the functionality is on par with GSR. The supply current and the instantaneous power drawn are also similar as seen in 4th row.

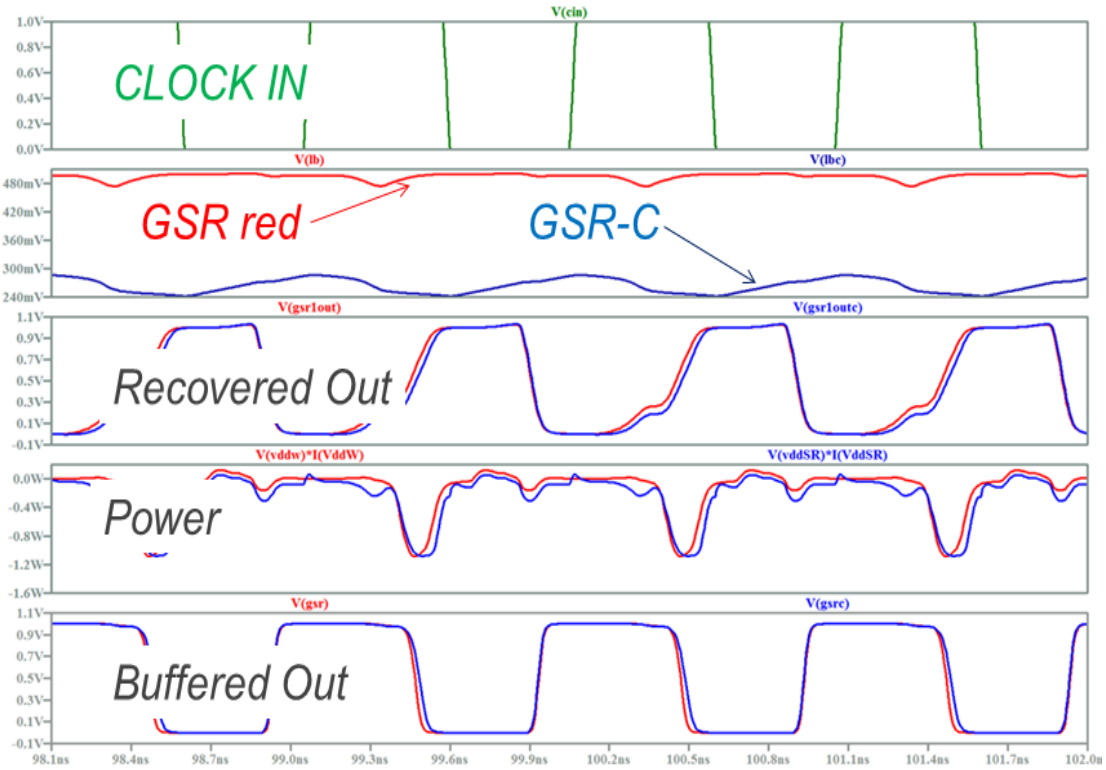


Figure 3.10 Simulations of GSR and GSR-C showing the functionality.

Operation at multiple voltages is shown in Figure 3.11, plotting the power drawn for driving a 20pF load in the functional frequency range for DVFS. Higher V_{DD} supply voltages give large frequency sweep but take higher power. Power is saved by moving to an operating point of lowest V_{DD} for a given frequency. No interconnect resistance is factored so that output swings rail-to-rail with a tank Q = 3. Lower supply voltages give lower maximum frequency but take less power at functional frequencies. The ability to scale voltage down to the minimum needed at any given frequency enables DVFS. The quadratic relation of power to V_{DD} explains

the spacing between the curves in Figure 3.11. The GSR simulated power at 1V and 1GHz is nearly half of $C_L V_{DD}^2 f_{CLK}$ as per (3.8). System level simulations with real life clock trees are shown in Chapter 9.

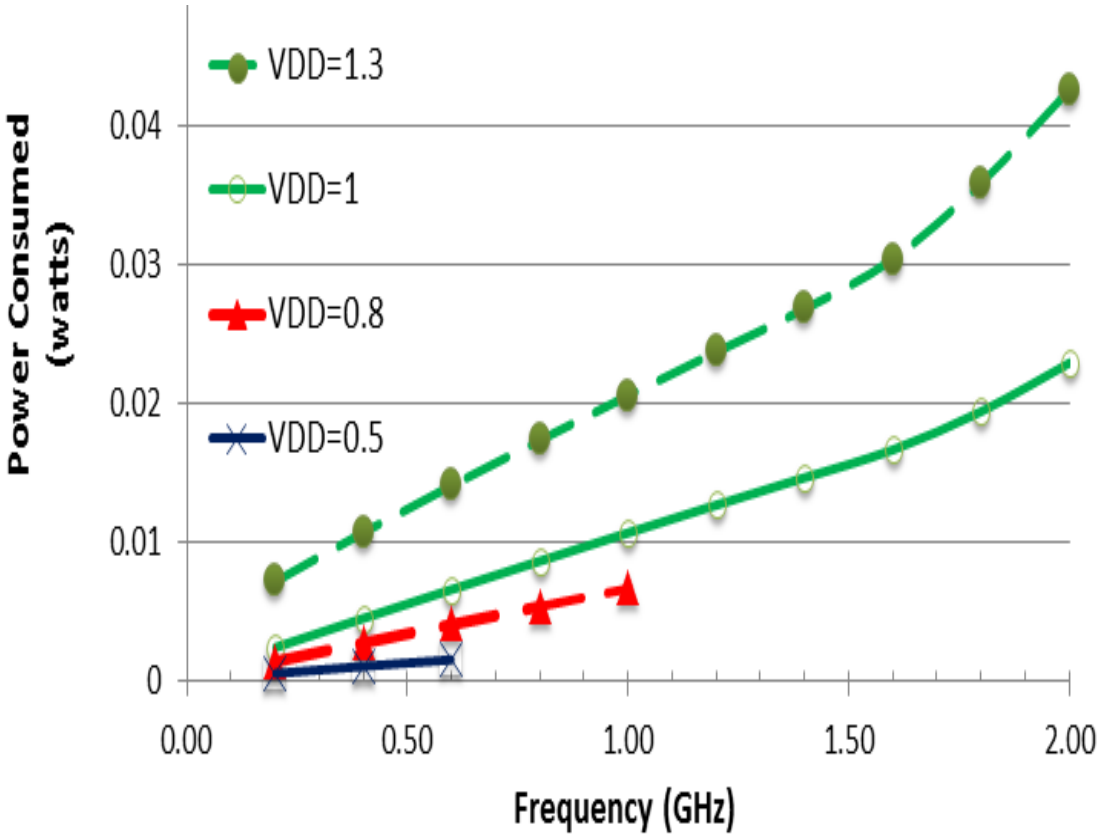


Figure 3.11 GSR Voltage and Frequency scaling operation for DVFS.

3.5.3 GSR Schematic Diagrams

Scalable CMOS implementation of GSR is shown in Figure 3.12. External connections of this macro cell determine the mode in which it is used. The macro symbol I shown at the bottom of Figure 3.12. In case of distributed inductance the transistor *Mr* will be placed outside this macro. The device widths will be scaled based on the technology's minimum channel length *L* used. The operation has been verified from 90nm to 22nm. The sizing of width also depends on the load capacitance driven. The sizes shown are for 1pF capacitance. For multiple pFs of capacitance, the parameter *sCL* scales the widths by having more devices in parallel.

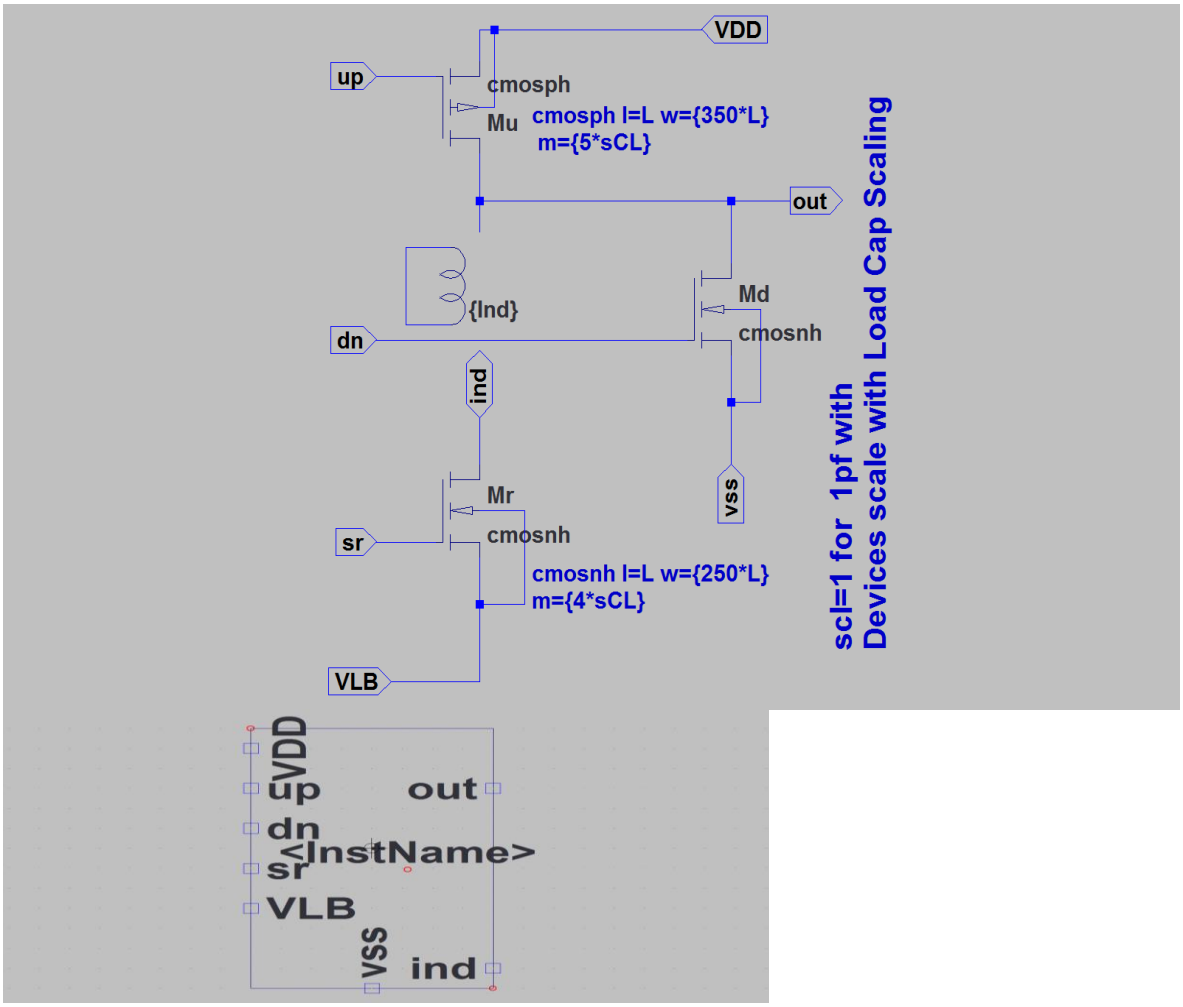


Figure 3.12 GSR Scalable Reconfigurable Driver Schematic and Macro Cell Symbol.

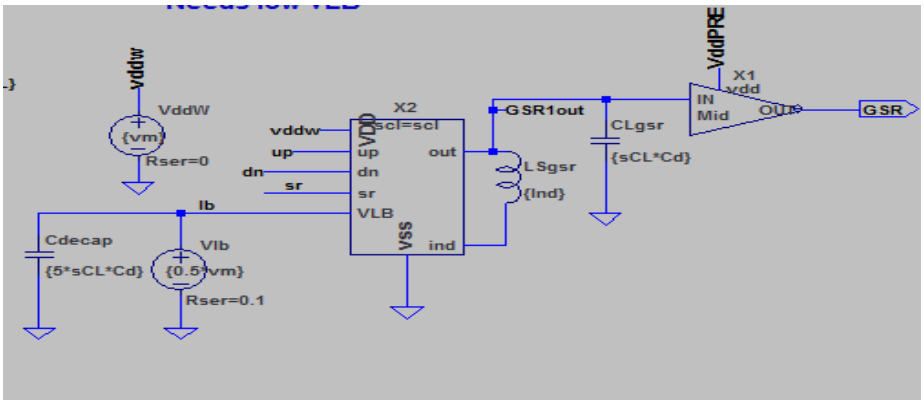


Figure 3.13 Typical Configuration of Driver for GSR rail to rail operation.

Figure 3.12 shows a typical GSR configuration for a load capacitance of sCL×1pF and inductor bias set to half the power supply used to generate the waveforms in Figure 3.10 using LTSPICE simulator.

4 SUPPORT CIRCUITRY

This chapter describes the transistor level implementation of the different blocks shown so far. It details the important circuits for realization of the complete GSR solution in practice and how they can be used in other configurations as well. Low power implementation of one or more of the following functions are needed for resonant and non-resonant operation:

1. Pulse Generators with controlled width
2. Multiple non-overlapping pulse streams
3. Voltage Doublers
4. Extra supply voltage $V_{DD}/2$ or bias generation

4.1 GSR Configuration

Figure 4.1 shows how the above 1, 2 and 3 may be realized. An optimum delay of $0.5T_R$ is generated from the *RLC* and inverter in the input stage of Figure 4.1. The series inductor (L_D) is a replica of L_S (from Figure 3.7), and matching capacitance C_{M1} tracks the load C_L. The pulse width, $0.5T_R \leq \pi\sqrt{L_S C_L}$ in Figure 4.1, is determined by $\pi\sqrt{L_D C_{M1}}$. The inductor L_{PW} is chosen large enough so that $T_{PW} = 2\pi\sqrt{L_{PW}(C_{Mr} + C_{M2})}$ is slightly larger than $0.5T_R$. Matched delays create pulse widths that are replica of load capacitance resonance times. GSR inductor control output is at double the supply voltage to reduce switch on-resistance. Here C_{Mr} is the non-negligible gate capacitance of the inductor switching transistor *Mr* in GSR scheme shown in Figure 3.7. C_{M2} is also matched to C_L like C_{M1}. This replica timing eliminates the need for synchronization with conventional DLL/PLL circuitry that would otherwise have required more area and power.

Repeated low going pulses are generated from both the edges of the input *CLOCKin* using an *XNOR* gate and the replica delayed signal. The *XNOR* output can be inverted to obtain the V_{SR} signal that controls the GSR inductor switch. The other two signals V_{UP} and V_{DN} are readily obtained through logical operations of *CLOCKin* and the *XNOR* output.

Thanks to the Miller gain around C_{M1} buffer, it is not necessary to have the entire load capacitance duplicated for replica delay. This saves power in charging and discharging this capacitor as well. For run-time tuning, accounting for inductor and load capacitance variations, the variable resistor R_{opt} can be tuned to adjust the *RLC* delay and change T_R appropriately. C_{M1} and C_{M2} can be varied to match the loads used, during die to die calibrations.

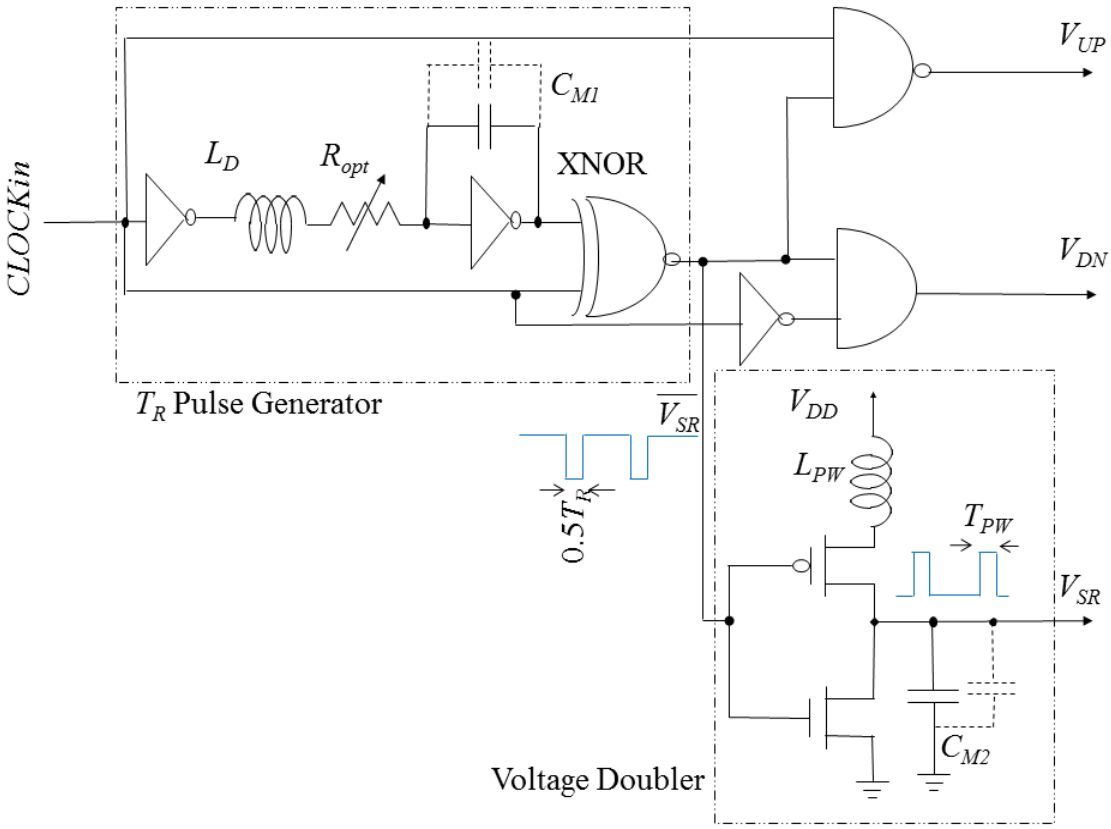


Figure 4.1 Generating control signals for GSR Driver.

The switch on resistance in GSR, for the same device size as NR, will be higher due to source bias voltage of $0.5V_{DD}$ in the NMOS. The drain source resistance is inversely proportional to gate source voltage V_{gs} and is given as $L/2\mu CoxW(V_{gs}\text{-}V_t)$ [38], [39]. While V_{gs} is full gate voltage of V_{DD} in NR case, in GSR it is only half that, as the source is now biased at $0.5V_{DD}$. Transistor width (W) can be increased to compensate for this but will increase area and capacitance. Other alternative is to drive the gate with higher voltage [24]. Resonant techniques can also be used to drive the V_{SR} line itself [40].

A low power voltage doubler scheme for V_{SR} is shown in Figure 4.1 that uses pulsed resonance technique. Pulse resonance based PMOS driver is used as a voltage doubler. The GSR inductor control output (V_{SR}) can swing at twice the supply voltage [15]. The circuit is actually a PMOS complement of PSR driver discussed. When the PMOS switch is closed, the inductor series resonates with the capacitance C_{M2} and C_{Mr}. The series inductor (L_{PW}) needs to be large enough to give the $0.5T_R$ timing needed at V_{SR}, with the additional load of GSR driver gate capacitance C_{Mr}.

For large load capacitances (>10pF) the resonant inductance values are quite small (<0.1nH) allowing the use of larger values of L_{PW} to give lower area C_{M2}.. For load capacitors a Q_C > 30 is assumed at 5GHz giving less than 1Ω of series resistance per 1pF. While the aspect ratio W/L is indeed large (> 600), resulting gate capacitance of 10fF increases the switching power of a 1pF load only by 1/33rd. The dominant GSR predriver capacitance is $2C_L$ for dynamic power calculations and can thus be effectively scaled to <0.2 C_L for large loads.

To estimate the power of this predriver, it can be seen equivalent to switching of 10 logic inverters, and capacitance $\leq 2C_L$ coming from input delay and doubler

output capacitance (that absorbs the gate capacitance of Mr switch $C_{Mr} < C_L\ /33$). With 5× Miller gain and 10× inductor value than the driver inductance value L_S, the effective capacitance driven can be < $0.2C_L$. Each logic inverter (termed INV) too has total input and output capacitance < $C_L/33$ across various processes from 90nm to 22nm. The minimum power predriver can this be estimated as,

$$P_{P\text{-}GSR} = \frac{10}{33} C_L V_{DD}^2 f_{CLK} + 0.2\ C_L V_{DD}^2 f_{CLK} \approx 0.5\ C_L V_{DD}^2 f_{CLK} \quad (4.1)$$

This is similar to NR overhead with tapered buffers. The signal generator of Figure 4.1 can be shared among 3 or more GSRs with the same T_R requirements to reduce power and area overhead to less than $0.2\ C_L V_{DD}^2 f_{CLK}$. The use of inductors in pre-drivers as well lowers the power needed to drive capacitive loads in the support circuitry while achieving the doubler function. While the doubled voltage means 4 times the power, the PSR structure reduces the power to 1/3rd.

The bias voltages needed by CPR, PSR and GSR are readily available in modern multi-voltage domain SoCs, especially in mobile processors. The $V_{DD}/2$ bias line draws no effective power as more current is pushed into it than pulled out. The output impedance requirement of this, as a fraction of total resistance R_T, can be calculated so that Q is not degraded to adversely affect the condition for underdamped oscillation and performance. For efficient energy savings, the output impedance of these is targeted to be less than 10% of the switch on-resistance.

4.2 PSR Reconfiguration and Application

PSR driver needs only a portion of the support circuits from GSR. It is well suited to drive level sensitive latches like true single phase latches (TSPC) [27]. A part of Figure 4.1 GSR pre-driver used for PSR is shown in Figure 4.2 along with data latches. Pulse Generator from GSR configured for PSR driver that clocks a bank of n

TSPC latches. The LC delay of pulse generator matches the resonance pulse width of PSR output. In the absence of the voltage doubler, inductor bias V_{LB} as low as $V_{DD}/4$ may be used, to achieve lower V_{OL} levels when effective Q value is very small. The pulse widths are programmed to full T_R rather than $0.5T_R$. The pulses are available on both edges of clock to support DDR.

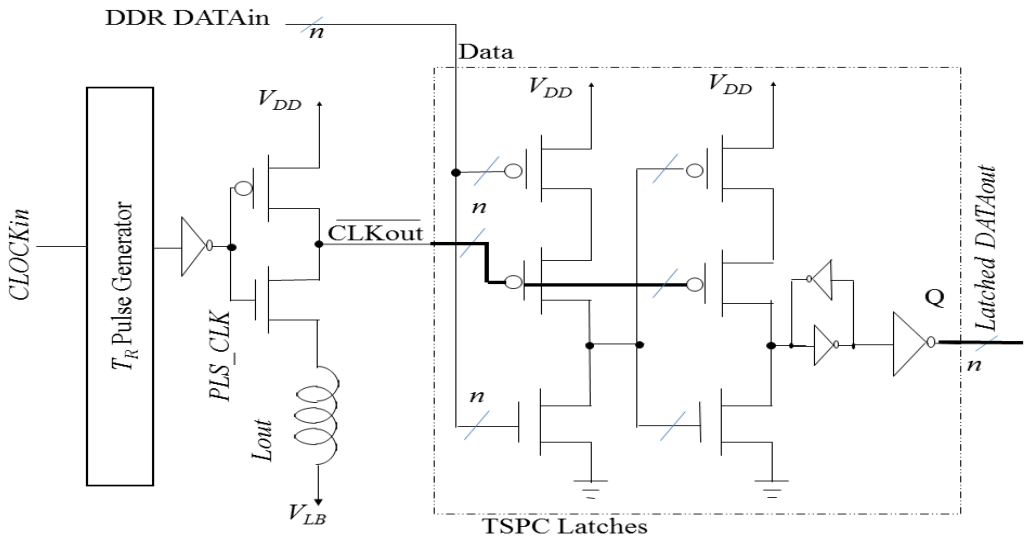


Figure 4.2 PSR driver clocking a bank of n TSPC latches.

To take advantage of the pulsed nature of the PSR driver output, the true single phased clocked latch (TSPC) shown in Figure 4.2 can be used instead of master-slave flip flops [23]. This latch is often called explicit-pulsed true single phase clocked flip flop (epTSPC) [21], [27], [31], [33]. TSPC latches also demand steep and controlled slopes of the enabling clock edge to prevent malfunctions from undefined values and race conditions.

The predriver portion of PSR takes roughly half the power of GSR giving,

$$P_{P\text{-}PSR} = \frac{5}{33} C_L V_{DD}^2 f_{CLK} + 0.1\, C_L V_{DD}^2 f_{CLK} \approx 0.25\, C_L V_{DD}^2 f_{CLK} \quad (4.2)$$

The predriver can be shared among 3 or more GSRs with the same T_R requirements to reduce power and area overhead to less than $0.1\, C_L V_{DD}^2 f_{CLK}$.

The PSR can create the controlled sharp falling edges needed to correctly trigger latches. The width T_{PW} needs to be large enough to complete one cycle of LC resonance and meet the latch transparency window target. PSR enables extra power savings in DDR applications. An ideal dual edge-triggered (DET) flip-flop allows the same data throughput as a single edge-triggered flip-flop while operating at half the clock frequency by sampling DDR. The power in the CDN is reduced by a factor of two or more if voltage is scaled as well.

PSR can achieve dual edge operation with TSPC latches without having to double the circuitry [23]. By clocking explicit-pulsed latches no additional flip-flop area is needed for double data rate operation [23]. This reduces the frequency and voltage for operation giving 40% area and power reductions for 1024 flops in 45nm CMOS process as shown in [23], [41]. All the required transistor level topologies to implement the solution have been shown in Appendix B: LTSPICE Schematic Diagrams

4.3 Flip-flops For Energy Recovery

The best ways to combine PSR with flip-flops to save local clocking power is now examined. Flip–flops are the basic elements of synchronous designs. Their choice and implementation can reduce the power consumption and provide more slack time for the timing budget. Various dual edge-triggered flip–flops compared in [7] have been extensively referenced and used [42], [43]. This includes implicit-pulsed flip–flops and explicit-pulsed flip–flops. Pulse-triggered flip–flops, characterized by a simple structure, negative setup time and soft edge, perform better than traditional master–slave flip–flops [43]. The pulse generator of the explicit-pulsed flip–flop can be shared by neighboring flip–flops, contributing to less power dissipation than implicit-pulsed flops. DET flip–flops can reduce the clock frequency

to half that of the single-edge flip–flop while maintaining the same data throughput, so that power dissipation is decreased [42]. These are reviewed now for use with P.

4.3.1 Conventional Solutions

At the leaf end of the tree, high-performance and low-power, energy recovery flip-flops that operate with resonant clocks have been proposed, exhibiting significant reduction in delay, power, and area [21], [31]. Another approach for energy recovery clocked flip-flops is to locally generate square-wave clocks from a sinusoidal clock. This technique has the advantage that existing square-wave flip-flops could be used with the energy recovery clock. However, extra energy is required in order to generate and possibly buffer the local square waves.

One of the lowest energy and area flip-flops reported in [21] is the Single-ended Conditional Capturing Energy Recovery (SCCER) flip-flop. This is representative of what are called implicit-pulsed dynamic flip-flops. It has differential circuitry to handle the special sine waves of CR drivers. With the PSR of Figure 4.2, these features may be redundant and so is the need to generate implicit pulses in every waveform. This pulse generator has the same function as the input stage T_R pulse generator in Figure 4.1.

Figure 4.3 from [43] shows an explicit-pulsed, hybrid semi-dynamic flop (ep-DCO) that consumes extra energy for the explicit pulse generator. However, this power consumption can be significantly reduced by sharing a single pulse generator among a group of flip-flops. Due to the dynamic nature of the circuit, back-to-back inverters are needed to hold the state of the intermediate output and the final output.

The ipDCO and epDCO with shared pulse generators are the best among all semi-dynamic flip-flops considered for use in high speed critical paths. The explicit-pulsed, hybrid static flip-flop (epSFF) from Intel in [43] is the most energy-efficient

of all the flops with time-borrowing (negative setup time) capability. The tradeoff is that the minimum delay of epSFF is larger than the minimum delay of epDCO. It is appropriate for the large number of paths on a chip which are speed-sensitive and can benefit from a fast delay and large amount of time-borrowing.

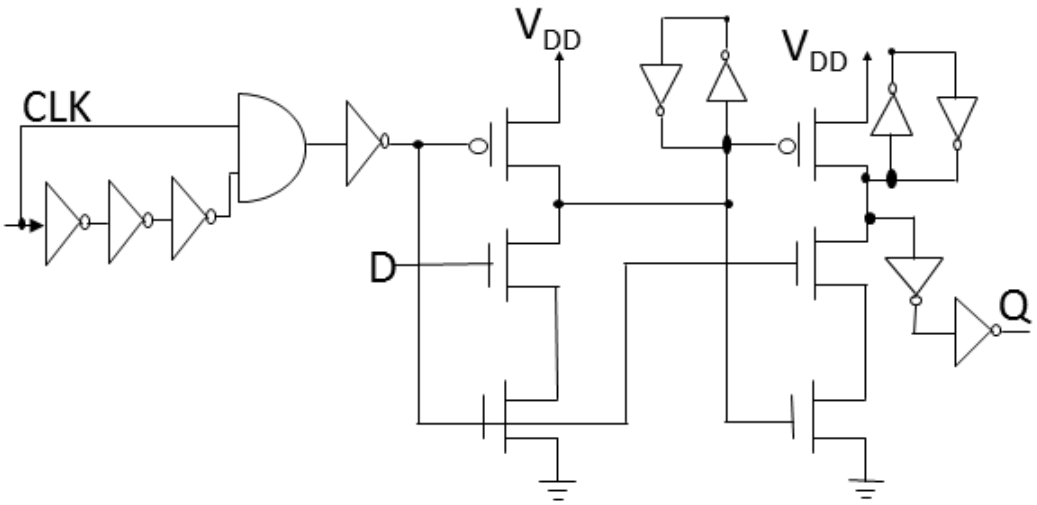


Figure 4.3 Explicit-pulsed flip-flop epDCO.

4.3.2 Dynamic Latch Solutions for PSR

The true single phased clocked latch (TSPC) is a compromise between the above two, with proven reliability, robustness and scaling advantages. Thus the choise is to pair TSPC with the explicit pulse output of PSR. This is the combination shown in Figure 4.4, termed as explicit-pulsed true single phase flip-flop (epTSPC). The main advantage is the use of a single clock phase. Dynamic output nodes are isolated by static inverters to prevent charge sharing effects.

Although simpler split output versions are possible, this topology allows for the targeted voltage scaling from 1.3V to 0.5V. Careful sizing on internal transistors is necessary to prevent glitching, even for static data [27]. TSPC latches also demand steep and controlled slopes of the enabling clock edge to prevent malfunctions from

undefined values and race conditions. As simulated before and described later, the PSR naturally creates the controlled sharp falling edges from resonance, to trigger correctly the bank of TSPC latches and interconnect (C_L).

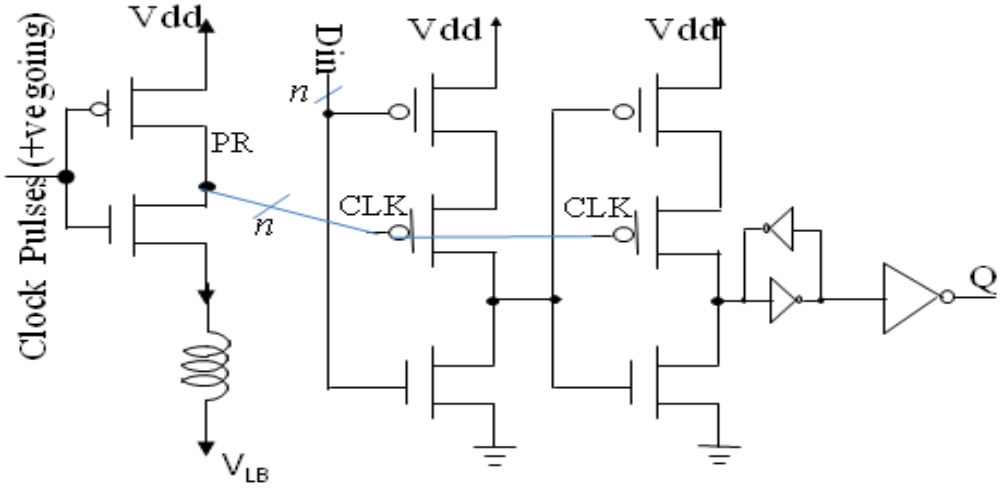


Figure 4.4 epTSPC driven by PSR.

An ideal dual edge-triggered (DET) flip-flop allows the same data throughput as a single edge-triggered flip-flop while operating at half the clock frequency and sampling data on both edges of the clock. If the clock load of the DET flip-flop is not significantly larger than the single edge-triggered version, the power in the clock distribution network is reduced by a factor of two. Dual edge operation for epTSPC simply implies that the explicit pulse generator gives pulses at both edges of the clock. The epTSPC of Figure 4.4 works on negative pulses from the PSR of Figure 3.2. For dual edge triggered TSPC (deTSPC), some of the circuit structure needs to be replicated with appropriate change in devices as shown in Figure 4.5. These are used with conventional clock drivers for power savings comparison. While epTSPC has lesser transistors, the burden falls on the PSR to have additional circuitry to generate controlled pulses on both edges of the incoming clock.

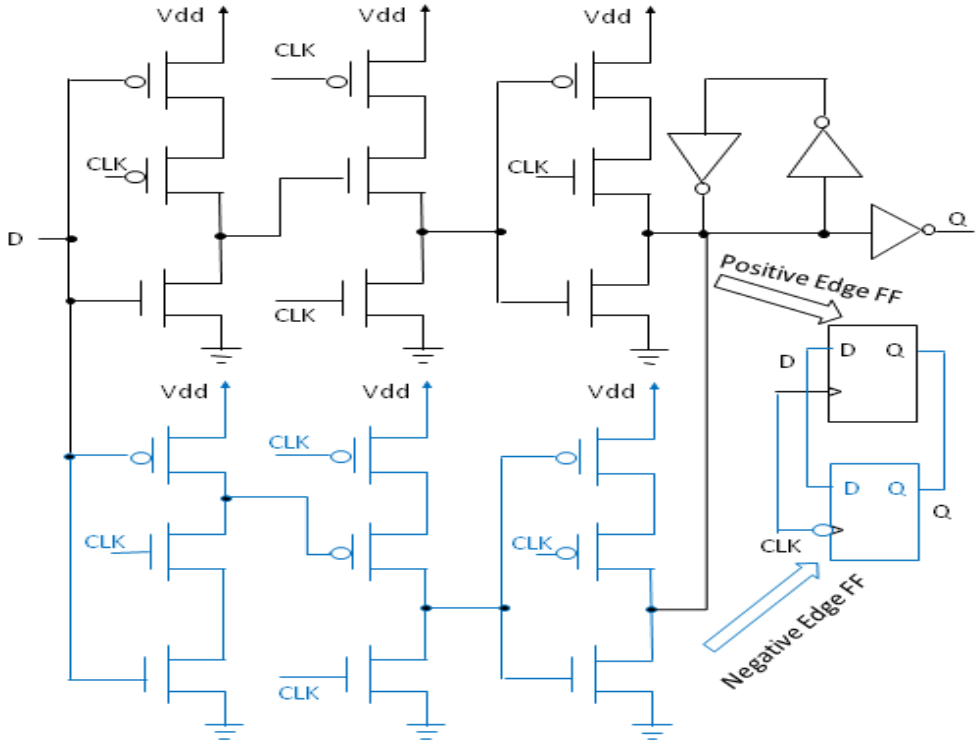


Figure 4.5 Dual Edge Triggered TSPC based Flip Flop (deTSPC).

4.4 PSR Flip-flop Functional Verification

Figure 4.6 compares the data capture edges with the clock leading data at both the rising and falling edges. NR with deTSPC fails to capture data with no set-up time.

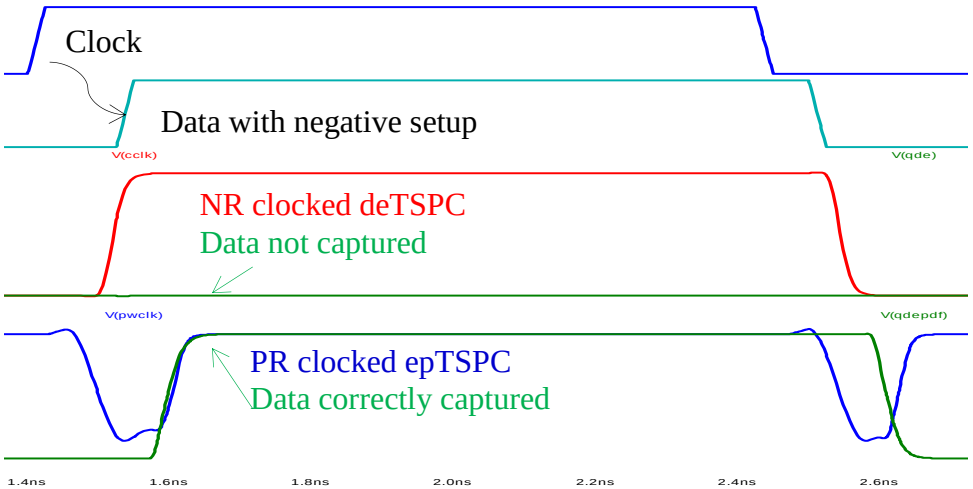


Figure 4.6 DeTSPC vs. epTSPC DET for negative setup.

PSR with epTSPC captures the data correctly even with the negative setup time. This can be used advantageously for clock de-skewing purposes. The hold time for epTSPC is well defined by the width of the resonance pulse and the clock to Q propagation (t_{dCQ}) is 4 inverter delays. Thus, the t_{dCQ} can be kept larger than hold time to minimize hold time violations for timing closures.

As an example of PSR, for a load of 1pF, a matching capacitance of less than 0.2pF is sufficient for generating 200ps pulses with 1nH inductor. These component value choices are made at design time. For run-time adjustments, the variable resistor R_{opt} can be tuned to adjust the *RLC* filter delay and minimize dynamic power. The matching mechanism from design time ensures functionality over PVT corners and mismatches. Run-time tuning is more energy efficient. The GSR system simulations are shown in Chapter 9.

4.5 GSR Functionality and Performance

The functionality and robustness of the new GSR driver and pre-driver circuitry is verified by 22nm SPICE simulations across 30% variation in *LC* component values and transistor model parameters [36], [37]. The results plotted in Figure 4.7 show that, in spite of some outliers, the GSR output V_C is functional to drive standard local buffers generating a *CLOCKout* signal for flip flops and other parts of the digital system. Signals from Figure 4.1 are shown to check robustness over 30% variations in values of active devices and passive components. Temperature is swept from -25^{o}C to 125^{o}C. Signals correspond to Figure 3.4 and a standard inverting buffer giving CLOCKout. The pulse width of V_{SR} varies to track the changes in the *LC* resonance time that come from variations in load capacitance. The pull up V_{UP} and pull down V_{DN} signals are always non-overlapping.

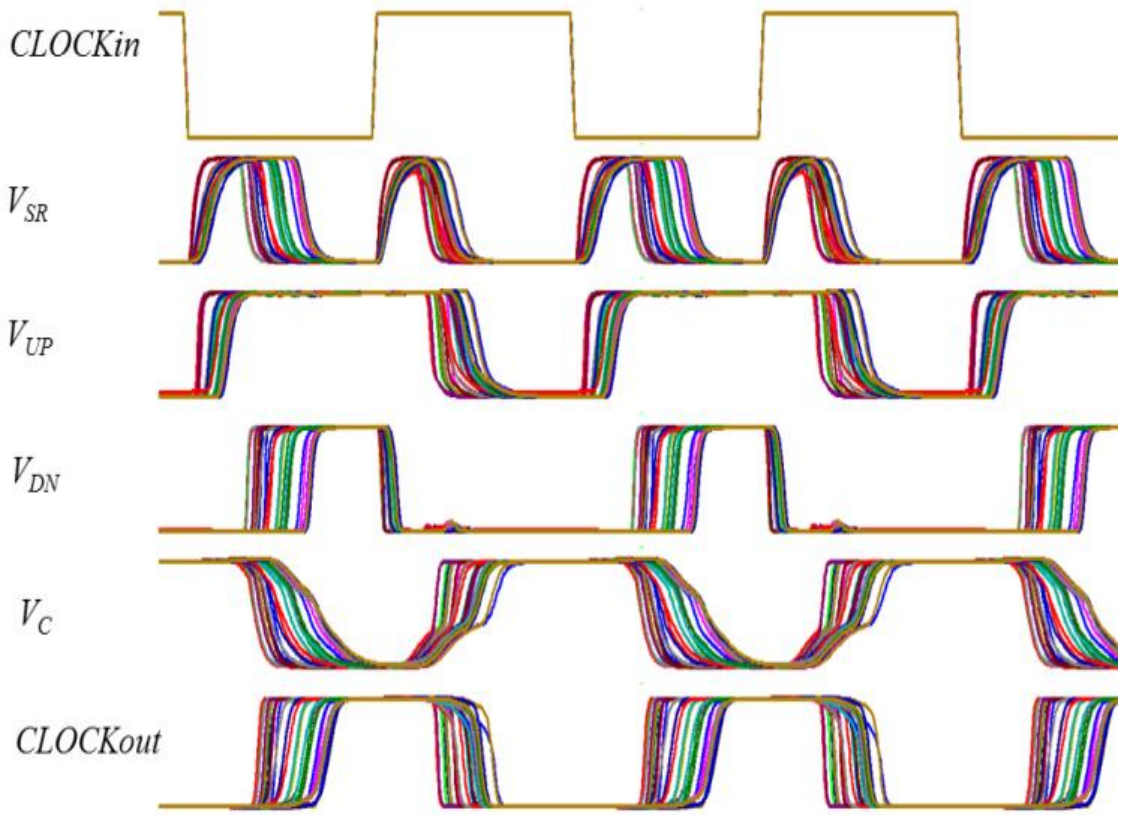


Figure 4.7 Monte Carlo simulations of GSR with predriver.

4.6 Circuit Design Optimizations

Figure 4.2 showed a novel PSR sub-system with an input delay generator for the required pulse width. The series input inductor with a Miller multiplier of matching capacitance generates an *LC* filter delay equal to one pulse width. This acts as a replica delay and tracks the PSR output resonance pulse width of T_R. The width needs to be large enough to complete one cycle of *LC* resonance as discussed earlier. The width is also chosen to meet the latch transparency window target. Thanks to the Miller gain, it is not necessary to have the entire load capacitance duplicated for replica delay. For a given load capacitance the feedback capacitance can be just 20% or less of the load capacitance to minimize area overhead.

Lower capacitance values can be used as well with higher series resistance. If more area is available, the entire replica of load capacitance can be connected from

resistor to ground instead of being across the inverter. This efficient circuit can drive epTSPC, meeting the requirements of robustness and controlled steep slew rates. The pulsed resonance, naturally creates the controlled sharp falling edges. The input stage that generates pulses can be shared among multiple PSRs if the T_R requirements are homogenous among the drivers.

It is possible to use the CPR and GSR drivers, replacing extra supply voltage of $V_{LB} = V_{DD}/2$, with a large bias capacitor C_B ($\approx 10 \times C_L$) charged to $0.5V_{DD}$, as shown in Sections 2.2.2 and 3.3. The operating equation is similar to (3.10) but power savings may be less. A pull up switch is required in CPR for this case. It takes several cycles for the output clock to be stable after turn on, so clock gating is not possible with this scheme [12], [14]. In GSR the total bias capacitance C_{ER} can be slightly smaller ($\approx 8 \times C_L$), to build and hold a bias voltage on the inductor storage end. The power consumption is similar to GSR, but with an extra factor $(1+C_L/C_B)$ in power equation (3.16). CPR, PSR and GSR described in Chapter sections 2.2.1, 3.1 and 3.2 do not lose any cycles in settling to the final waveform and thus can be clock gated.

5 TIMING PERFORMANCE OF DRIVER SOLUTIONS

Skew and jitter are very critical performance parameters as they directly affect timing closure at high speeds in the nanometer regime, taking significant design time resources. Slow slew rates affect skew and jitter, as well as cause short circuit currents. As insertion delays are used to match timing skews, large variations in propagation delay are also detrimental to achieving closure over process variations. Based on the circuit models and output voltage equations derived in Chapter 2 and Chapter 3, the propagation delays and slew rates of the various clock drivers discussed so far are now analyzed. The intrinsic gate delays are ignored as the C_L is assumed to be much larger than device capacitances.

5.1 Propagations Delays and Transition Times

Propagations delay t_{PD} is the delay from the mid-rail of the input to the mid-rail of the output. Transition times are the output rise and fall times between the 90% V_{DD} and 10%V_{DD} points. The slew rate is the slope of transitions at mid-rail.

5.1.1 Non-Resonant Driver

The delay to midpoint, averaging over rise and fall, can be obtained from (1.1) as shown in [27] as,

$$t_{PD} = \ln(2)\,[R_w+(R_u+R_d)/2]C_L = 0.69[R_w+(R_u+R_d)/2]C_L. \quad (5.1)$$

This propagation delay does not include any predriver delay. To minimize overall delay tapered buffers are used as predrivers in practice [20], [28]. Tapered buffer for minimum delay have excess capacitance that converges to $C_L(\frac{1}{n-1})$, where n is the number of buffers. When n=3 the excess capacitance from predrivers is $0.5C_L$. Accordingly the excess power in NR predriver is given by,

$$P_{P\text{-}NR} = = 0.5\, C_L V_{DD}^2 f_{CLK} \quad (5.2)$$

The criteria for minimum delay implies that the delay in each stage is the same [27]. Thus, the total insertion delay through the predrivers and drivers is given by,

$$t_{\mathrm{INS}} = (n+1)\, t_{\mathrm{PD}} = 0.69\,(n+1)[R_w + (R_u + R_d)/2]C_L. \tag{5.3}$$

From (1.1), the 90% to 10%V_{DD} fall time can be calculated as [27],

$$T_{\mathrm{fall}} = 2.2(R_d + R_w)C_{L.} = T_{\mathrm{rise}} \tag{5.4}$$

The rise time is identical as it is governed by a similar equation. This is usually kept less than 10% of the clock period. An upper bound reduces the effect on setup/hold constraints and decreases short-circuit power. A lower bound is also needed to reduce peak supply currents and cross-coupling noise and electromagnetic interference (EMI).

Skew between two clock lines can occur due mismatch in routing lengths and variation input threshold of the buffers due to device process variations and supply/signal voltage differences. The equivalent offset voltage ΔV of the buffers is proportional to supply voltage V_{DD} by a proportionality ε, giving $\Delta V = \pm\varepsilon V_{DD}$. The slew rate *SR* at the input can be calculated from (1.1) as,

$$SR_{NR} = \frac{-V_{DD}}{(R_d + R_w)C_L}.e^{\frac{-t}{(R_{d/u}+R_w)C_L}} \tag{5.5}$$

The worst case skews t_{skw} in the buffer clock lines, assuming $+\Delta V$ offset on one buffer and $-\Delta V$ offset on the other can be estimated as below at V_{DD} /2 using (5.*1*) as,

$$\begin{aligned} t_{\mathrm{skw}} = \frac{2\Delta V}{SR_{NR}} &= 2\frac{\varepsilon\, V_{DD}(R_{d/u} + R_w)C_L}{V_{DD}}.e^{\frac{t}{(R_{d/u}+R_w)C_L}} \\ &= 4\varepsilon\,(R_{d/u} + R_w)C_L \end{aligned} \tag{5.6}$$

The value of ε is typically contained to be less than 10% through matching so that skew t_{skw} is more than half of the driver propagation delay t_{PD}. Clock skew is typically budgeted to be 10% the minimum time period T_{CLK} (at maximum operating frequency) so that rest of the timing budget can be allocated to logic path delays and setup/hold times. Since logic designers anyway consider this in their timing constraints, further reduction at the expense of power is unnecessary. This implies that the propagation delay t_{PD} of final NR driving long times needs to be less than 20% of the clock period, assuming that the predrivers are shared and only contribute to the total insertion delay.

The supply/ground variations and cross-talk from other signals can be taken as changing threshold level in the buffers and causing the variation in delay from input to output arrival. Assuming peak-to-peak variation the power supply is $\Delta V = \pm\beta V_{DD}$ with other cross-talks are combined into this, the peak-to-peak time variation based on slew rates can be derived similar (5.6) to as,

$$t_{\mathrm{jit-pp}} = \frac{2\Delta V}{SR_{NR}} = 4\varepsilon\beta\,(R_{d/u} + R_w)C_L \qquad (5.7)$$

Again, supply variations can be contained within 10% by careful shielding, decoupling and limiting of current spikes [39]. This would give $t_{\mathrm{jit-pp}}$ to be only 1/10$^{\mathrm{th}}$ of t_{skw}, the skew for each buffer which is less than 1% of T_{CLK}. However, the jitter is accumulated over the entire buffer clock buffer chain inserted and not just the final buffer. This chain may have upto 10-20 buffers. The peak-to-peak jitter values from each buffer do not directly add but the standard deviations of the variation in the Gaussian distribution can be summed. The final jitter number will depend on the number of buffers used, as is the insertion delay, and typically comes to be same order

as the skew t_{skw}. Clock jitter can often dominate the timing budgets and insertion delays are often minimized to handle this.

Larger device sizes can decrease switching resistance and reduce basic delays. Wider interconnect lines can minimize the skew and jitter, but all at the expense of more power. NR has the lowest delays and transit times and fully supports DVFS but takes higher power than other drivers described below.

5.1.2 Continuous Parallel Resonance (CPR)

CPR waveform given by (2.5) takes a quarter of a cycle to reach the midpoint voltage at resonance frequency, resulting in a propagation delay of $T_{RES}/4$. Combining with underdamped condition of $L_p < 4R_p^2 C_L$, delay of CPR for this maximum allowed inductance can be approximated for high Q as,

$$t_{\mathrm{PD}} \leq T_{RES}/4 = 2\pi\sqrt{4R_p^2\, C_L C_L}/4 = \pi R_p\, C_L \tag{5.8}$$

This is larger than NR case as $Rp > (R_u+R_d)/2+R_w$. Less delay with smaller R_p implies smaller Q and will have less energy saving efficiency as per (2.6). Thus, one sees a tradeoff between delay and energy across driver circuit topologies.

Buffer sizes needed for CPR are much smaller than for NR and thus no significant predriver is needed. Tapered buffers are not necessary and the excess capacitance can be kept less than 1/10$^{\text{th}}$ of NR at $0.05C_L$. Thus the insertion delay t_{INS} is nearly same as the propagation delay t_{PD}. Excess power in predriver is thus only,

$$\mathrm{P_{P\text{-}CPR}} = = \frac{1}{2}\, C_L V_{DD}^2 f_{CLK} \tag{5.9}$$

For CPR the 10% - 90% rise time (fall time) points of the sinusoidal signal in (2.5), can be shown as [33],

$$T_{\mathrm{rise/fall}} = 0.29 T_{CLK}. \tag{5.10}$$

This is nearly 30% of the clock period rather than the desired 10%.When the rise times are long, as is the case for low frequencies, it leads to power and delay performance degradation. This is one of the reasons CPR is still not widely adopted.

Skew and jitter can be derived similar to NR case with slew rate derived from differentiating (2.5) and evaluating at $T_{RES}/4$ as,

$$SR_{CPR} = \frac{V_{DD}}{2} 2\pi f_{RES} \sin(2\pi f_{RES} T_{RES}/4) = \pi V_{DD} f_{RES} \tag{5.11}$$

The skew can then be derived from (5.5) as below,

$$t_{\text{skw}} = \frac{2\Delta V}{SR_{CPR}} = 2\frac{\varepsilon\ V_{DD}}{\pi V_{DD} f_{RES}} = \frac{2\varepsilon}{\pi}\ T_{RES} \tag{5.12}$$

For the same values of ε as NR, the skew from the driver itself is less than 7%, though the rise/fall transitions are 30%.

The final skew to be considered includes the delay mismatch from interconnects, and this can be large in CPR as the t_{PD} delay itself is larger. Overall, the skew tends to be larger for CPR. However, jitter is less as long buffer chains are avoided and most of it comes from the final driver itself, which can be derived similar to NR as below,

$$t_{\text{jit-pp}} = \frac{2\Delta V}{\pi V_{DD} f_{RES}} = \frac{2\varepsilon\beta}{\pi}\ T_{RES} \tag{5.13}$$

For the same ε and β as NR above, CPR has less than 1% of period as jitter from above, which is very beneficial. CPR also avoids EMI and jitter coming from multiple harmonics of clocks as the resonance operation by definition rejects all harmonics above the fundamental frequency. The problem of course is that, for clock frequencies away from the resonance frequency, the power increases non-linearly and DVFS is not supported.

5.1.3 Pulsed Series Resonance (PSR)

Unlike CPR, the resonance frequency in PSR can be higher than clock frequency, as $T_{RES} = 2\pi\sqrt{L_S C_L}$ is less than T_{CLK}. The propagation delay to $V_{DD}/2$ of the falling edge in Figure 3.2 is less than $T_R/4$. For large Q, this can be taken as $T_{RES}/4$. As keeping T_{RES} small implies using smaller inductors with higher Q, it is attractive to use PSR. Combining with underdamped condition needing minimum inductance as $L_S > R_T^2 C_L/4$, the delay relation for PSR can be approximated for high Q as,

$$t_{\mathrm{PD}} \geq 2\pi\sqrt{R_T^2\, C_L\, C_L/4}/4 > \pi R_T\, C_L/4. \tag{5.14}$$

However, as R_T is usually smaller than R_p, (5.14) can give an smaller delay value for PSR than CPR or NR. For more accurate results, T_{RES} can be replaced by $T_R = T_{RES}/\sqrt{1-\frac{1}{4Q^2}}$ from (3.4).

Predriver for PSR shown in Figure 4.2 shows the delay of T_{RES} and propagation delay of approximately three unit buffers given by (5.1). Thus the insertion delay t_{INS} is more than five times propagation delay t_{PD}, as shown by,

$$\begin{aligned} t_{\mathrm{INS}} &\approx T_R + T_R/4 + 3 \times 0.69\, [(R_u + R_d)/2]\, \frac{1}{33} C_L \\ &\approx \frac{1.25\pi}{\sqrt{1-\frac{1}{4Q^2}}} R_T C_L + \frac{1}{33}(Ru + Rd) C_L \end{aligned} \tag{5.15}$$

Similar to the sinusoidal waveform of CPR, the fall time from 90% to 10% points for PSR can be obtained from (3.5) as,

$$T_{\mathrm{rise}} = 0.29 T_R. \tag{5.16}$$

T_{rise} is larger than T_{fall} for lower Q (<14) as it includes the RC based pull up time shown in Figure 3.2(b). T_{fall} is only 6% of the clock period at the fastest rate, as series resonance frequency is typically set to at least five times the maximum clock frequency. Thus it is better than the desired 10% and well controlled over the entire

operation. When the rise times are small, even in the case for low frequencies, it leads to lower power from short-circuit currents. This is one of the advantages of PSR over CPR and NR.

Skew and jitter can be derived similar to NR/CPR case with slew rate derived from differentiating (3.5) and evaluating for high Q at t = T_{RES}/4 as,

$$SR_{PSR} = \frac{V_{DD}}{2} 2\pi f_{RES} e^{-tR_T/2L_S} \sin(2\pi f_{RES} t) = \pi V_{DD} f_{RES} e^{-\pi/4Q} \tag{5.17}$$

The skew can then derived from (5.5) as below,

$$t_{\text{skw}} = \frac{2\Delta V}{SR_{PSR}} = 2\frac{\varepsilon\ V_{DD}}{\pi V_{DD} f_{RES}} e^{\frac{\pi}{4Q}} = \frac{2\varepsilon e^{\frac{\pi}{4Q}}}{\pi} T_{RES} \leq \frac{2.5\varepsilon}{\pi}\ T_R \quad for\ \ Q \geq \pi \tag{5.18}$$

For the same values of ε as NR and CPR, the skew from the driver itself is less than 5% assuming $T_R < \frac{T_{CLK}}{5}$, since the series resonance frequency is typically 5× the maximum f_{CLK}. This is in addition to the timing budget savings from the rise/fall transitions. The final skew to be considered includes the delay mismatch from interconnects, and this is small in PSR as the t_{PD} delay itself is small. This is not counting the common predriver delay. The skew tends to be also frequency independent.

Jitter is less as long as buffer chains are avoided. Most of it comes from the final driver and predriver itself, which can be derived, similar to CPR as below,

$$t_{\text{jit-pp}} = \frac{2\Delta V}{\pi V_{DD} f_{RES}} e^{\frac{\pi}{4Q}} = \frac{2\varepsilon\beta e^{\frac{\pi}{4Q}}}{\pi} T_{RES} \leq \frac{2.5\varepsilon\beta}{\pi}\ T_R \quad for\ \ Q \geq \pi \tag{5.19}$$

$$\leq \frac{\varepsilon\beta}{2\pi} T_{CLK} \quad for \ Q \geq \pi$$

For the same ε and β as in NR and CPR, PSR has peak-to-peak jitter less than 1% of period T_{CLK}, which is very beneficial. The predrivers and three buffers add 1% each to give less 2% total jitter. PSR like CPR also avoids EMI and jitter coming from multiple harmonics of clocks by the virtue of its resonance.

Another advantage over CPR is that the switch closure time T_R set by *LC* resonance frequency is independent of the clock period T_{CLK}. This gives the wide frequency operation feature of PSR, down to the lowest clocking frequency. PSR does not have the problem of CPR in supporting DVFS. Additionally, the slew rate is set by the faster T_R time rather than the variable T_{CLK}. It is optimal to use PSR with level sensitive latches that only depend on controlled fall time. The pulse mode of operation can also save power downstream by replacing flip-flops with lower power latches [23], [24].

5.1.4 Generalized Series Resonance (GSR)

GSR has the same advantages over CPR as PSR. The delay equations remain the same but the fall time is faster with extra pull down switch. With multiple timing signals, GSR can give rail-to-rail outputs and 50% duty cycle outputs. The ability to interface with standard logic makes it more attractive to use than PSR or CPR. It takes more area for the extra switches and needs more support circuitry discussed in Section 4.1. GSR is a general purpose resonant scheme that can be reconfigured as PSR or CPR as shown in Section 3.4.

The delay equation for the driver alone $t_{\mathrm{PD}} \geq \pi R_T C_L/4$ is a valid approximation for GSR as well. The insertion delay from predriver of GFSR is

different from PSR predriver due an additional series resonance doubler stage embedded, giving the overall value as,

$$t_{\mathrm{INS}} \approx T_R/4 + T_R + T_R/4 + 3 \times 0.69\,[(R_u+R_d)/2]\,\frac{1}{33}C_L$$
$$\approx \frac{1.5\pi}{\sqrt{1-\frac{1}{4Q^2}}} R_T C_L + \frac{1}{33}(Ru + Rd)C_L \qquad (5.20)$$

This equation is good for comparative analysis but it is based on simplified linear models assuming a fixed load capacitance. The actual values will be different due to voltage dependent non-linear capacitances.

The slew rate is governed by the same equation (5.17) as PSR and can be taken as $SR_{GSR} = \pi V_{DD} f_R e^{-\pi/4Q}$ without loss of generality. The skew from the driver alone can then be bound by the relation $t_{\mathrm{skw}} \leq \frac{2.5\varepsilon}{\pi}\, T_R$ for $Q \geq \pi$.

For the same values of ε as NR and CPR, the skew from the driver itself is less than 5%, like PSR, as the series resonance frequency is typically 5× the maximum f_{CLK}. The final skew to be considered includes the delay mismatch from interconnects, and this is small in GSR too as the t_{PD} delay itself is kept small. This is not counting the common predriver delay.

The skew tends to be also frequency independent. Jitter is less as long as buffer chains are avoided. Most of it comes from the final driver and predriver itself, which can be derived similar to PSR as,

$t_{\mathrm{jit-pp}} \leq \frac{\varepsilon\beta}{2\pi}\, T_{CLK}$ for $Q \geq \pi$.

For the same ε and β as NR, CPR and PSR, GSR driver by itself has peak-to-peak jitter less than 1% of the clock period T_{CLK}. The predriver is a PSR stage having a jitter of 2% as discussed above. The overall peak-to-peak jitter then is less than

2.3% of $T_{CLK.}$ GSR like PSR/CPR also avoids EMI and jitter coming from multiple harmonics of clocks by the virtue of its resonance.

5.2 Comparative Analysis

The timing for GSR is compared with NR and CPR in Figure 5.1. PSR has similar results to GSR. The waveforms compared are from (1.1), (2.9) and (3.5) with the simulated delays and transition times for a 20pF load and <3Ω of switch resistance without any interconnect parasitics. The simulated delay values are within 10% of the theoretical calculations using (5.1) - (5.16). The pre-driver delays are not factored for simplicity as they do not affect slew rates appreciably.

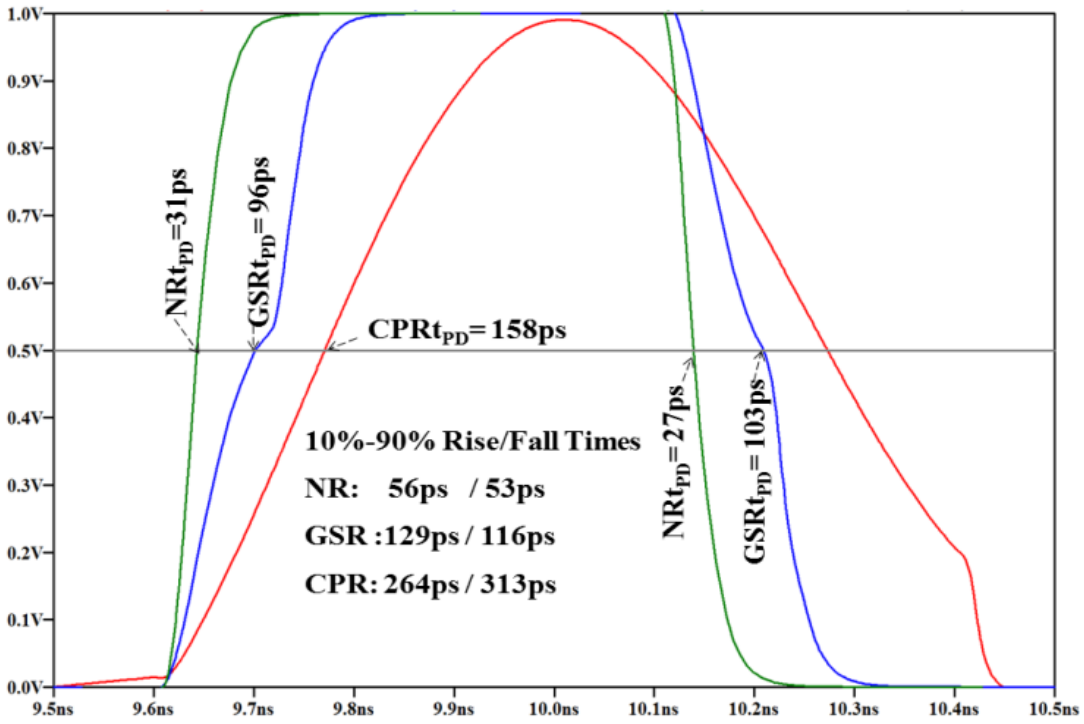


Figure 5.1 Simulated output voltage waveform on a 20pF load capacitor (V_C).

The resonant frequency of CPR is 1.0GHz. Propagation delays *(t_{PD})* to mid-points at 50% marker are shown vertically on individual curves. The NR curve is the fastest with maximum pull up and down strengths. With the same sizes, CPR launches

a rising sinusoidal wave, whose falling edge does not need a triggering input. Thus no t_{PD} is shown for falling edge of CPR. GSR has smaller delays than CPR.

6 DATA PATH APPLICATIONS

While special latches were used for power savings with PSR clock, it is desirable to have logic blocks for computation with lower power than standard CMOS logic. One such promising logic family is the domino-style dynamic logic that is traditionally encumbered by the clocking power. While several innovations like [44] have helped the use of dynamic logic in mainstream for higher speed at lower power, they have not shown energy recycling advantage presented here. By using the GSR principles in the clocking of standard dynamic logic, power can be saved in the switching required in every cycle irrespective of data. The refresh cycle of domino logic naturally performs the pull up function in the GSR. Thus the GSR predriver can readily generate the clocking signals for dynamic logic operation.

6.1 Resonant Dynamic Logic (RDL)

In dynamic logic gates, the output is pulled to V_{DD} during refresh/pre-charge phase of the clock cycle T_{CLK} [44]. Valid input is required only during the evaluation phase of the period. Figure 6.1 shows a resonant version of domino-style dynamic logic [45]. Figure 6.2 shows the timing signals necessary for the correct logical operation of RDL. While the pre-charge (REF) and evaluate (EVAL) signals are also part of the resonant gate operation shown below, an additional phase is needed for energy recovery with the timing signal REC. When input IN is at logic 1, the inductor is disconnected from the output. When IN is at logic 0 it is connected to the output twice before the next clock cycle starts. M1 functions as the refresh switch. M2 is used to charge and discharge capacitor C through inductor. The preprocessing CMOS gate shown will generate the necessary control voltages to connect and disconnect the inductor to save and recover energy. The EVAL and REC active low pulse widths are

$0.5T_{LC}$ for resonance operation. T_{LC} like T_{RES} before is given by $2\pi\sqrt{LC}$ and is a fraction of T_{CLK} in order to fit two units of it in the Evaluate and Recover phases. The logic expression for L_{ON} is given by, L_{ON} = EVAL. $\overline{IN}$ + REC. $\overline{OUT}$

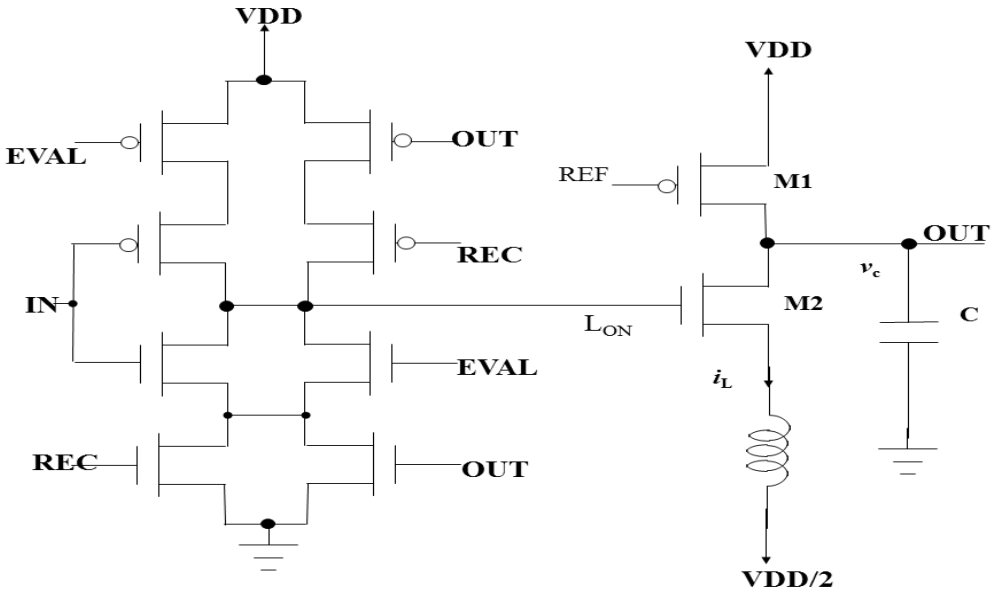


Figure 6.1 CMOS Implementation of Resonant Dynamic Logic (RDL).

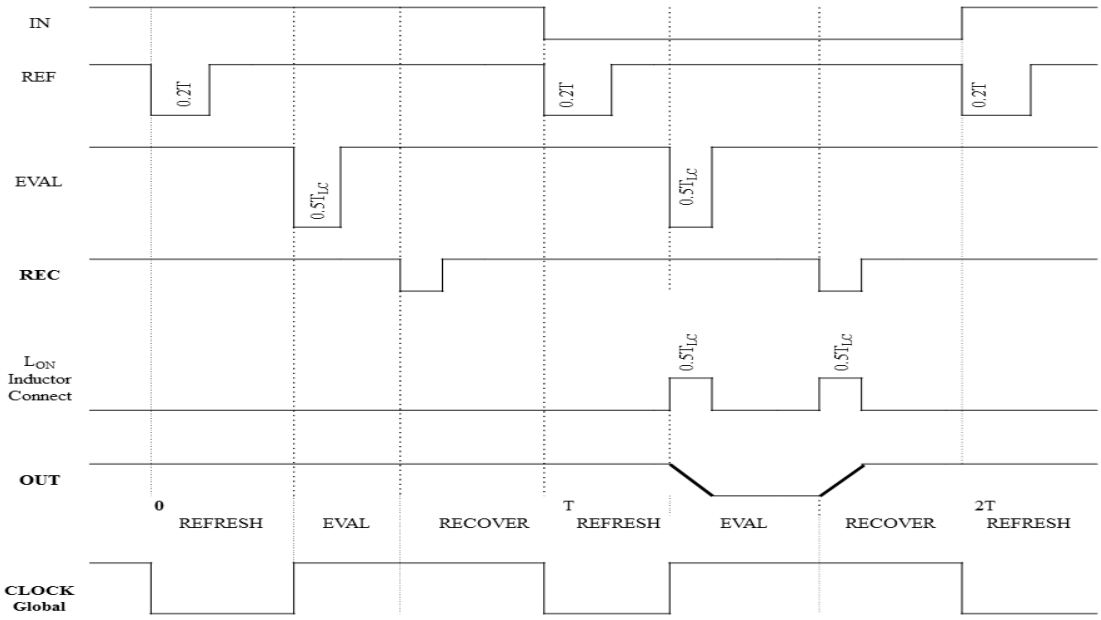


Figure 6.2 Timing signals derived from clock supporting energy recovery switching.

At the end of the recovery, the refresh switch M1 is momentarily closed by REF pulse to compensate for finite Q losses and bring OUT voltage fully back to V_{dd}. The refresh switch may also be closed during logic 1 to account for any charge leakage from the capacitor. Note that the inductor is only utilized during the transition times and otherwise free for rest of the cycle.

For input IN = 0, L_{ON} is high and M2 connects the inductor to the output load capacitor C. By lossless resonance given by (6), OUT goes to ground when the switch is closed for duration (T_{LON}) of $0.5T_{LC}$. Thus the correct logical evaluation for the driver with the energy stored in the inductor supply is achieved. For the OUT = 0 now, L_{ON} evaluates to high (V_{DD}) again with active low REC pulse for L_{ON}. The M2 switch is again closed for another short period of $0.5T_{LC}$. This will restore the output to the pre-charge value V_{DD}, assuming ideal lossless transfer of energy from the inductor supply to output load capacitor. To compensate for finite *Q* losses, the refresh switch M1 is momentarily closed by REF pulse, at the end of the recovery, to bring the voltage fully back to V_{DD}. This is the operation during which most power is consumed. The governing equation is identical to (3.8) derived for PSR as 0.5 $(1\text{-}e^{-\pi/Q})$ $V_{DD}^2\ C_L f_{CLK}$. Figure 6.1 shows the preprocessing logic for a simple inverter, but it can be extended for an *n* input logic gate driving appreciable line capacitance *C*.

6.2 RDL Power and Delay

The NR power for static logic is given by standard expression, with data switching at most half the clock rate, with an activity factor α as,

$$P_{Static} = \frac{1}{2}\alpha C V_{DD}^2 f_{CLK} + \frac{n}{33}\alpha C V_{DD}^2 f_{CLK} \qquad (6.1)$$

The second term accounts for the *n*-input logic processing. Activity factor indicates the fraction of times that the output signal goes high. For NR dynamic logic power is only consumed on low going signals but at twice the rate as signals are pulled high immediately after being pulled low. This would give the power for an *n*-bit domino style dynamic logic as,

$$P_{\text{Domino}} = (1-\alpha)CV_{DD}^2 f_{CLK} + + \frac{n+1}{33}\alpha CV_{DD}^2 f_{CLK} \quad (6.2)$$

This includes the second term for the extra power for the *n* input logic preprocessing combined with the clock. Thus, while dynamic logic can give fastest data rates and smallest propagation delays possible for a given clock, it does not give the lowest power possible for any data rate as the data is toggled on the high capacitance output node like a clock. In fact the power is almost double for an even case of α=0.5.

For RDL using the power savings from (3.8) of the PSR structure, the total power can be estimated for comparative analysis as,

$$P_{\text{RDL}} = \frac{1}{2}(1-\alpha)(1-e^{-\pi/Q})\, CV_{DD}^2 f_{CLK} + \frac{n+1}{33}\alpha CV_{DD}^2 f_{CLK} \quad (6.3)$$

In comparison for α=0.5, and a realizable $Q \geq \pi$, RDL power is a third of standard domino logic power and 50% less than standard static logic. Thus the advantages of dynamic logic's fastest processing are realized without the power penalty, by using RDL. This of course more practical for large size *C* that would make the necessary inductor *L* value small enough. The propagation delay for fall time t_{PD}, can be derived similar to PSR as,

$$t_{\text{PDR}} \leq T_R/4 + 3 \times 0.69\, (2\, Ru)\, \frac{n+1}{33}\, C_L \quad (6.4)$$

$$\leq \frac{1.25\pi}{\sqrt{1-\frac{1}{4Q^2}}} R_T C_L + \frac{4(n+1)}{33} RuC_L$$

This is of course larger than standard domino NMOS only delays, but can still be kept less than the delay of standard CMOS logic.

6.3 RDL simulations

The W/L ratio for M2 is kept large enough to minimize the ON resistance and to maximize the effective quality factor (Q) of the LC tank. The charge/discharge time $0.5T_{LC}$ is a fraction of the main clock period set at 0.2 T_{CLK}. The inductor needed is less than 5nH for a 1pF load at 1GHz for T_{LC} =0.4ns.

Figure 6.3 shows simulation results using BSIM3 models for a 90nm standard CMOS MOSIS process. An on-chip capacitor is assumed as the load, that is equivalent to driving 800 unit area (.1µ x .1µ) transistors for clock/data lines or 2mm long interconnects. Power is compared a non-resonant (NR) domino style circuit driving same load..

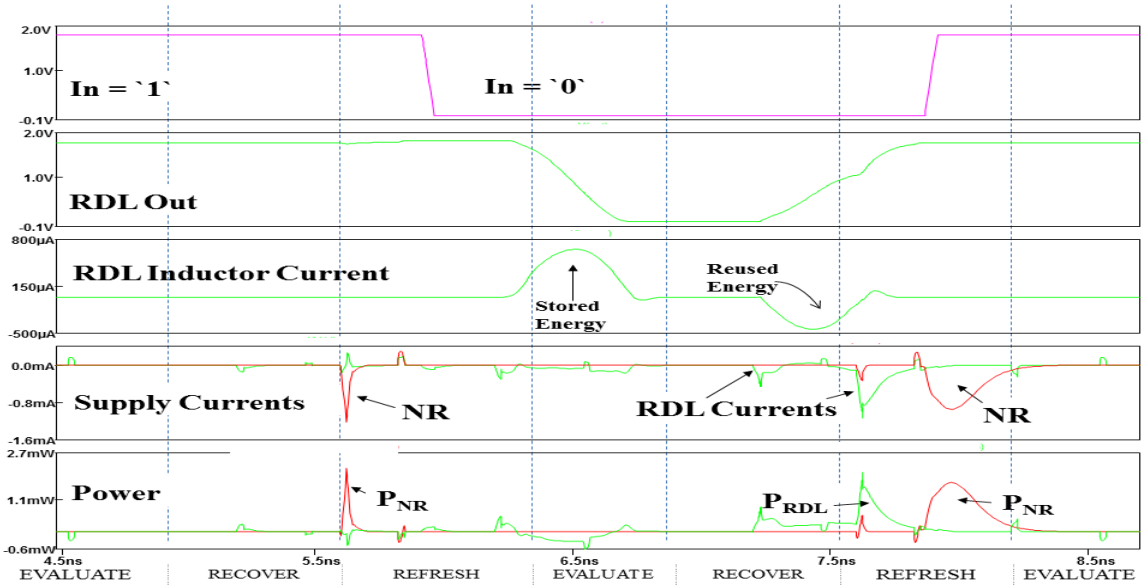


Figure 6.3 Operation at 1.8V supply and 0.5GHz.

Simulation results show that at 0.5GHz rate they match well with the theoretical description of the resonant operation. The output voltage discharges in the evaluate cycle for IN=0, and charges up again in recover phase. The inductor current curve in Figure 6.3 shows the sinusoidal operation. An on-chip value of about 3 is targeted for the *Q* factor.

When the inductor switches off, a certain amount of overshoot or ringing may be seen in the inductor current at a higher frequency. This is due to parasitic capacitances and the residual energy left in the inductor. While a smaller *Q* actually helps in reducing the ringing, it will also diminish the power savings. Keeping the switch closed for a slightly longer time helps to recover extra energy and can give more power savings. Note that the inductor is only utilized during the transitions times and is otherwise free for rest of the cycle. The same inductor may thus be shared among different logic cells using dual phase clocks.

7 AREA ESTIMATES

This chapter deals with the layout and area considerations that affect the performance of clock drivers and the distribution. Clock skew is directly related to the clock tree and other interconnect topologies chosen. There are several choices to be made when trying to minimize mismatches and variations while keeping the power to a minimum. There is also the concern for the area of inductors and their proper placement. Though inductors in theory do not take active area but just metal, excessive metal usage can cause routing blockages and directly inhibit timing closure. Placement of inductors close to supply lines can cause eddy currents that cut down the value of inductance and quality factor as well. Thus the layout is an integral part of design, just as in analog design, when it comes to CDNs using inductors for resonant recycling of energy. In general it is assumed that reasonable increase in area is acceptable when reducing power consumption. The common H-tree is shown in Figure 7.1.

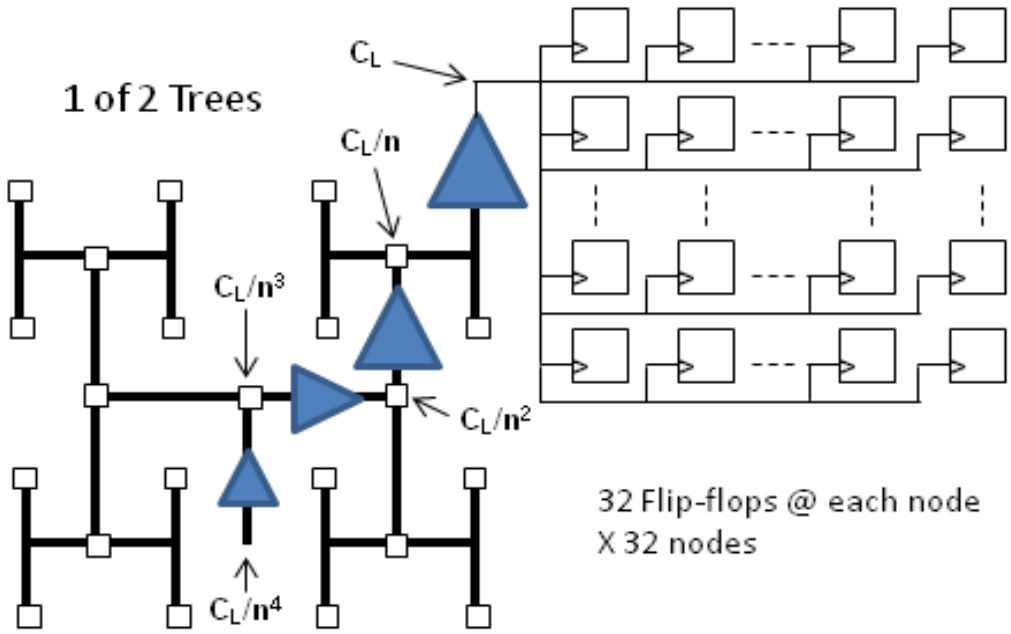


Figure 7.1 Distributed Clock Tree driving 1024 flip-flops.

The H-tree distributes the clock signal in a symmetric fashion to multiple sinks with minimum mismatch and skew. The implementation of a complete clock and data sub-system in the SoC is shown in [28] with scalable tapered buffers, also known as driver horn. This is used as a benchmark CDN in this thesis to compare the power dissipations and performance in Chapter 8.

The total input capacitance for the local bank of flip-flops and the connecting wires shown as C_L, may not be identical for each branch of the tree. The gain *'**n**'* is balanced evenly across the driver stages with the input capacitance of each stage being the output capacitance divided by *'**n**'*. Figure 7.1 represents the actual implementation of a 4-stage tapered buffering shown at the bottom of Figure 1.2 for NR clocking. Resonant clocks do not need to use such horns and directly drive the local C_L loads. But they can have the inductors distributed along the horn or at the far end.

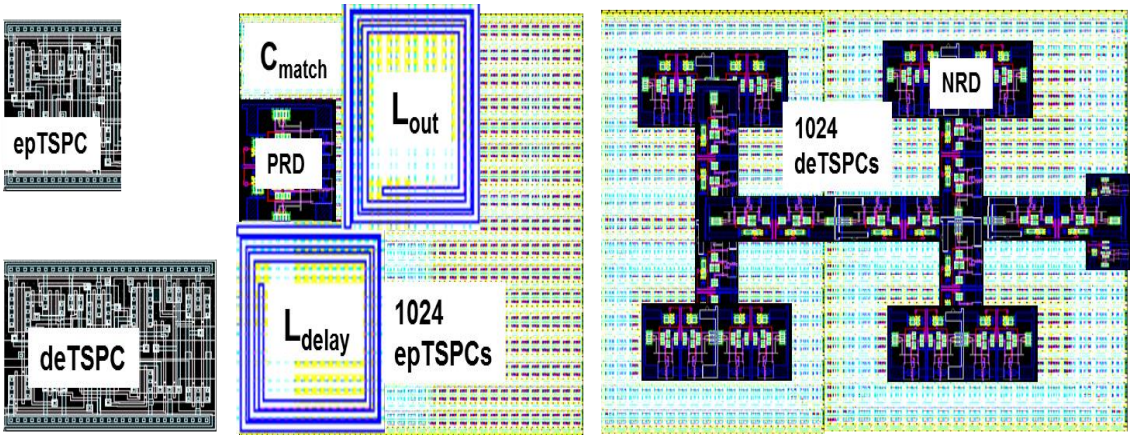
All the driver schemes shown need additional circuitry for input pulse stream generation. NR and GSR need non-overlapping pulses. CPR needs a minimum timing pulse width for a given driver size for proper operation **[**34]. Keeping the pulse widths minimum will minimize the static leakage in large driver devices. The predriver requirements are also important in determining total power and silicon area.

7.1 PSR Implementation in 45nm

The area of the PSR output stage is equivalent to 5 medium-sized standard inverters (INVs) which have a 10μm NMOS and 14.6μm PMOS in the IBM/PTM 45nm technology [32]. The rest of the active circuitry shown in PSR predriver takes the equivalent of 6 INVs. In contrast NR buffer horn as represented in Figure 7.1 would take 64 such INVs. Thus there is a 4× reduction in active area with PSR. The clock is distributed using an H-tree network on a metal layer with wires of 0.1Ω/ μm

resistance and 0.2fF/ μm capacitance. Clock skew can be reduced by wires in parallel at the expense of more power. With proper sizing and spacing of clock wires, the clock skew targets can be met [18]. Figure 4.1 replaces the entire chain of 64 inverters driving the clock tree.

The layout plan of these cells is shown in Figure 7.2 as verified in Calibre. The epTSPC takes less than 60% of deTSPC area as illustrated in Figure 7.2 (a) in a cell to cell comparison of epTSPC vs deTSPC. The flips-flops, grouped into 32x32 registers, are distributed across 100μm x 100μm in Figure 7.2 (b). Complete PSR with predriver and the 1024 epTSPCs can fit in the 100μm x 100μm area shown in Figure 7.2 (b). PSR driver PRD includes the predriver.



(a) epTSPC vs deTSPC (b) PSR driver PRD 1024 esTSPCs (c) NR Driver NRD and 1024 deTSPCs

Figure 7.2 Layout Floor Plan for comparing PSR and NR Clocking.

Two 1nH inductors, needed for PSR and its predriver, can be best implemented in the top metal layer well within the 100μm x100μm area above the active area of the flops. The deTSPC flips-flops, grouped into 32x32 registers, are distributed across 100μm x 100μm in Figure 7.2(c). Additional 50% area is needed for NR buffer horns as shown in Figure 7.2(c). The Non Resonant Driver NRD drives 1024 deTSPCs for minimum delay. Thus PSR clocking takes 40% less area

than NR as the 1024 deTSPC flops alone take 10,000μm^2 area and 50% more is needed for NR buffer horns.

The complete leaf cell test bench of 1024 flip-flops clocked by PSR through an H-tree clocking network can be extracted. The extracted parasitics from layout affecting the performance are used in SPICE simulations.

7.2 GSR Implementation

GSR can also be implemented in similar fashion to PSR. The flip flop area will be similar to NR. The predriver of GSR takes roughly twice the size of PSR. When driving entire clock tree loads (>100pF), the matched capacitors in GSR predriver of Figure 4.1 can take excessive area. Making the inductors L_D and L_{PW} 10 times or more can scale the capacitance area down by 10×. Inductors' extra metal area is not usually considered as they can be stacked on top of the active area of the predriver.

The GSR predriver takes an equivalent of only 16 INVs compared to 6 for PSR. However, NR driver does need predrivers (nearly 5 INVs) to reduce delays in driving the large gate capacitance of clock drivers leading to tapered buffers [20]. In an NR H-tree clock distribution, the extra capacitance driven can be 50% of C_L for optimal delays, leading to 50% more power [20]. CPR buffer sizes are small compared to other schemes. For comparison, NR needs 8 INVs to drive a load of 1pF with optimal delays; CPR takes less than 4 INVs; PSR takes 5 INVs and GSR 15.

The inductor value for a given resonance frequency and capacitance is given by $1/4\pi^2 f_{RES}^2 C_L$. For nominal load capacitance values of 1pF, an L_P of more than 25nH is needed for CPR at clock speed of 1GHz. In GSR/PSR, giving some margin for pull up/down time, the resonance width ($T_R=1/f_R$) is usually set at about 1/5th of nominal

T_{CLK}, resulting in 5× larger value for resonance frequency than the clock [24]. The series inductor value is then smaller, given by $L_S=L_P(f_{CLK}/f_R)^2$. For the 1pF load at 1GHz clock rate, T_R can be set to 0.2ns using a 1nH inductor resulting in a 5GHz f_R. Both PSR and GSR need less metal area for inductors in the driver compared to CPR. Inductor metal area for PSR and GSR can be on top of the driver active area and not encroach on other active areas as shown in Figure 7.2. The inductor metal usage can sometimes affect critical performance due to routing blockages in the clock tree synthesis. PSR can also use bond wire inductors or off-chip inductors, especially for low frequency operation [24]. In GSR implementation, distributed coils have the transistor *Mr* distributed at multiple locations as well, as shown in Figure 7.3.

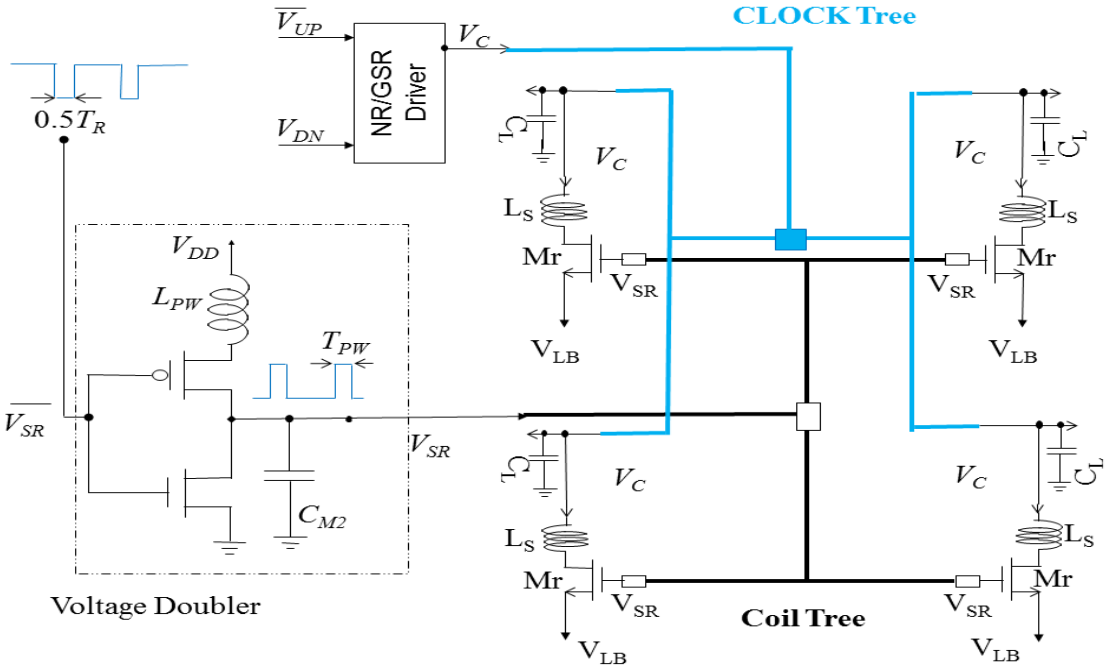


Figure 7.3 GSR distributed at far-end for highest Q and minimum power.

The extra capacitance of distributing V_{SR} is already factored in C_{M2} tuning capacitance shown. This can be as high as the load C_L when distributed to far-end, next to the load. For multiple parallel lines, the widths can be decreased to lower the capacitance at the expense of series resistance. In addition, inductor sizes can be

decreased to handle larger capacitance values. Optimal inductor location whether it is at the driver end, or far end, or at middle, involves trade-offs in power and skew. It is to be noted that the split-NR driver topology, commonly used, also needs two clock distribution lines one for NMOS and the other for PMOS.

7.3 Inductors

The inductor layout can be quite involved to achieve best possible quality factors as reported in [11]. As many as a 1000 inductors can be used in a processor design. In CPR their sizes can be prohibitive enough to block normal place and route. For series resonance schemes this is not an issue.

8 PERFORMANCE POWER AREA (PPA) TRADE OFF ANALYSIS

In order to facilitate Performance Power and Area (PPA) analysis, all the information from previous chapters has been summarized in Table 1. It shows a transistor level implementation of GSR topology and reconfigurations for NR, CPR or PSR operation. When all devices are used, GSR operation is enabled with correct control signals as in Figure 3.4. Mode may be selected for the best performance or power at the frequency of operation. As an example, for low frequency wafer testing NR may be used. For DVFS, GSR or PSR may be used. For maximum clock speed and savings at a single frequency CPR may be optimal.

CPR gives the lowest power if the resonance frequency can be set at or below the operating frequency. Parallel LC resonant circuit operation can operate in sinusoidal mode with reduced buffer sizes. This is because the on-resistance *Ru* or *Rd* being higher does not adversely affect power consumption, as long as the oscillations are underdamped. If the device sizes are made smaller for CPR, the on-resistances will be higher, but *Rp* determining the delay in (5.8) is not directly affected. This reduces the pre-driver overhead to drive load capacitance C_L and lowers the total power further. Since only losses need to be overcome at resonance, after the initial start-up, additional power savings can be realized by reducing the strength of the clock buffers driving the *LC* load [3], [12], [14], [20]. More than 40% of power saving is predicted with optimal synthesis algorithms [3], [18]. In practice continuous resonant solutions with *L* always connected in parallel to *C* are shown to save 25% power or more [11]-[14], [19], [20].

Table 1 PERFORMANCE POWER AREA TRADEOFFS

(Bold text** indicates the configuration giving the best performance metric**)

	Non Resonance (NR)	**Continuous Parallel Resonance (CPR)**	**Pulsed Series Resonance (PSR)**	**Generalized Series Resonance (GSR)**
Application & Key Feature	Low Frequency Testing Smallest Delays	Fixed f_{CLK} for Global CDN Lowest power at high f_{CLK}	Pulse mode DDR Latches Lowest Power DVFS	**General Purpose Low Power. Driving standard gates.**
$V_C(t)$ Voltage on Load Capcitor (Ignoring capacitor ESR for all cases)	$\boldsymbol{V_{DD}.e^{\frac{-t}{(R_d+R_w)C_L}}}$	$\frac{V_{DD}}{2} - \frac{V_{DD}}{2}e^{-t/2R_pC_L}[\cos(2\pi f_R t) - \frac{1}{2Q}\sin(2\pi f_R t)]$	$\frac{V_{DD}}{2} + \frac{V_{DD}}{2}e^{-tR_T/2L_S}[\cos(2\pi f_R t) - \frac{1}{2Q}\sin(2\pi f_R t)]$	$\frac{V_{DD}}{2} + \frac{V_{DD}}{2}e^{-tR_T/2L_S}[\cos(2\pi f_R t) - \frac{1}{2Q}\sin(2\pi f_R t)]$
		$f_R = \frac{1}{2\pi}\sqrt{\frac{1}{LpC_L}(1-\frac{1}{4Q^2})}$ $R_P=(Q_L^2+1)r_S$, $L_P= L_S(Q_L^2+1)/Q_L^2$ Tank $Q_{CPR} = R_p/\sqrt{L_p/C_L}$	$f_R = 1/T_R = \frac{1}{2\pi}\sqrt{\frac{1}{LpC_L}(1-\frac{1}{4Q^2})}$ Tank $Q_{PSR} = \sqrt{L_S/C_L}/R_T$	$f_R = 1/T_R = \frac{1}{2\pi}\sqrt{\frac{1}{LpC_L}(1-\frac{1}{4Q^2})}$ **Tank $Q_{GSR} = \sqrt{L_S/C_L}/R_T$**

	Non Resonance (NR)	**Cont. Parallel Resonance (CPR)**	**Pulsed Series Resonance (PSR)**	**(GSR)**
Driver Power	$C_L V_{DD}^2 f_{CLK}$	$\frac{\pi}{4Q} C_L V_{DD}^2 f_{RES} + \frac{1}{Q} C_L V_{DD}^2 f_{CLK} - \frac{1}{Q} C_L V_{DD}^2 f_{CLK} \cos^2(\pi \frac{f_{RES}}{f_{CLK}})$	$0.5\ (1\text{-}e^{-\pi/Q})\ C_L V_{DD}^2 f_{RES}$	$(1\text{-}0.5e^{-\pi/2Q}\ \text{-}0.5e^{-\pi/Q})$ $C_L V_{DD}^2 f_{RES}$
		$Q_{CPR} \approx Q_L = R_P/2\pi f$ $L_P = 2\pi f L_S / r_S$	$Q_{PSR} = 2\pi f L_S/(R_r+R_W+r_S) < Q_L$	$Q_{GSR} = 2\pi f L_S/(R_r+r_S) < Q_L$
Driver Area	Proportional to C_L and Routing Lengths	<0.25 NR Active Area Large Inductor metal Area	Active Area ≈ NR Ind. Metal area < CPR	Active Area ≈ 1.25 NR Ind. Metal area < CPR
Predriver Capacitor & Inductor Overhead	$\leq 0.5 C_L$ & (n/a)	$< \mathbf{0.05} \boldsymbol{C_L}$, & (n/a)	C_L & L_S or $0.1\times C_L$ & $10\times L_S$	$2C_L$ & $2L_S$ or $0.2\times C_L$ & $20\times L_S$
Predriver Power (P_P) for n stages	$\leq 0.5 C_L V_{DD}^2 f_{CLK}$ n ≥ 3 & for min. delay	$< \mathbf{0.05} \boldsymbol{C_L V_{DD}^2 f_{CLK}}$	$\approx 0.1\ C_L V_{DD}^2 f_{CLK}$ *shared over ≥4 drivers*	$\approx 0.2 C_L V_{DD}^2 f_{CLK}$ *shared over ≥4 drivers*
(P_D + P_P) Total Power for $Q > \pi$	$< 1.5 C_L V_{DD}^2 f_{CLK}$	$> (\mathbf{\frac{1}{4} + 0.05}) C_L V_{DD}^2 f_{RES}$	$\approx (\frac{1}{3} + 0.1) C_L V_{DD}^2 f_{CLK}$	$\approx (\frac{1}{2} + 0.2) C_L V_{DD}^2 f_{CLK}$

	Non Resonance (NR)	**Cont. Parallel Resonance (CPR)**	**Pulsed Series Resonance (PSR)**	**(GSR)**
Driver Delay	0.69 $R_{NR}C_L$ $R_{NR}=(R_u+R_d)/2+R_w$ $R_{NR} < R_T < R_p$	$< \pi R_p\, C_L$ $R_p > R_T > R_{NR}$	$> \pi R_{TPSR}C_L/4$ $R_{TPSR} = (R_r + R_W + r_S)$ $R_{NR} < R_{TPSR} < R_p$	$> \pi R_{TGSR}C_L/4$ $R_{TGSR} = (R_r + r_S)$ $R_{NR} < R_{TGSR} < R_{TPSR} < R_p$
Predriver Delay	n×0.69 $R_{NR}\, C_L$	(n-1)×**0.69 $R_{NR}\, C_L$**	T_R + 0.69 $R_{NR}\, C_L$	T_R + 3×0.69 $R_{NR}\, C_L$
Insertion Delay	0.69 (n +1) $[R_w+(R_u+R_d)/2]C_L$.	$< \pi R_p\, C_L$	$\approx \frac{1.25\pi}{\sqrt{1-\frac{1}{4Q^2}}} R_T C_L +$ $(Ru + Rd)C_L$	$\approx \frac{1.5\pi}{\sqrt{1-\frac{1}{4Q^2}}} R_T C_L +$ $(Ru + Rd)C_L$
rise/fall times	2.2×$(R_{u/d}+R_w)C_L$	$0.29T_{CLK}$ @ $f_{CLK} = f_{RES}$ $(R_u+R_w).C_L << T_R < T_{CLK}$	$0.29T_R$ $(R_u+R_w).C_L << T_R < T_{CLK}$	$0.29T_R$ $(R_u+R_w).C_L << T_R < T_{CLK}$
Slew Rate	$\frac{V_{DD}}{(R_d + R_w)C_L} . e^{\frac{-t}{(R_{d/u}+R_w)C_L}}$	$\pi V_{DD} f_{RES}$	$\pi V_{DD} f_{RES} e^{-\pi/4Q}$	$\pi V_{DD} f_R e^{-\pi/4Q}$
Skew	$4\varepsilon\ (R_{d/u} + R_w)C_L$	$\frac{2\varepsilon}{\pi}\ T_{RES}$	$\leq \frac{2.5\varepsilon}{\pi}\ T_R \quad for\ \ Q \geq \pi$	$t_{skw} \leq \frac{2.5\varepsilon}{\pi}\ T_R$ for $Q \geq \pi$
Jitter	$4\varepsilon\beta\ (R_{d/u} + R_w)C_L$	$\frac{2\varepsilon\beta}{\pi}\ T_{RES}$	$\leq \frac{\varepsilon\beta}{2\pi}\ T_{CLK} \quad for\ \ Q \geq \pi$	$\leq \frac{\varepsilon\beta}{2\pi}\ T_{CLK} \quad for\ \ Q \geq \pi$

8.1 Tradeoffs between NR, CPR, PSR and GSR

As shown in Table 1, following are the pros and cons in choosing a scheme for a given application [23].

8.1.1 Power and Dynamic Voltage Scaling

Energy in all driver cases goes as the square of supply voltage as given by $C_L V_{DD}^2$. Plotted on a logarithmic scale, this would present a straight line for all drivers with same slopes but different offsets, as shown in Figure 8.1. At higher voltages though, the on resistance of switches is smaller, leading to larger tank Q and more energy savings for resonant schemes. This can be seen at higher voltages where the curves are below the linear extrapolation.

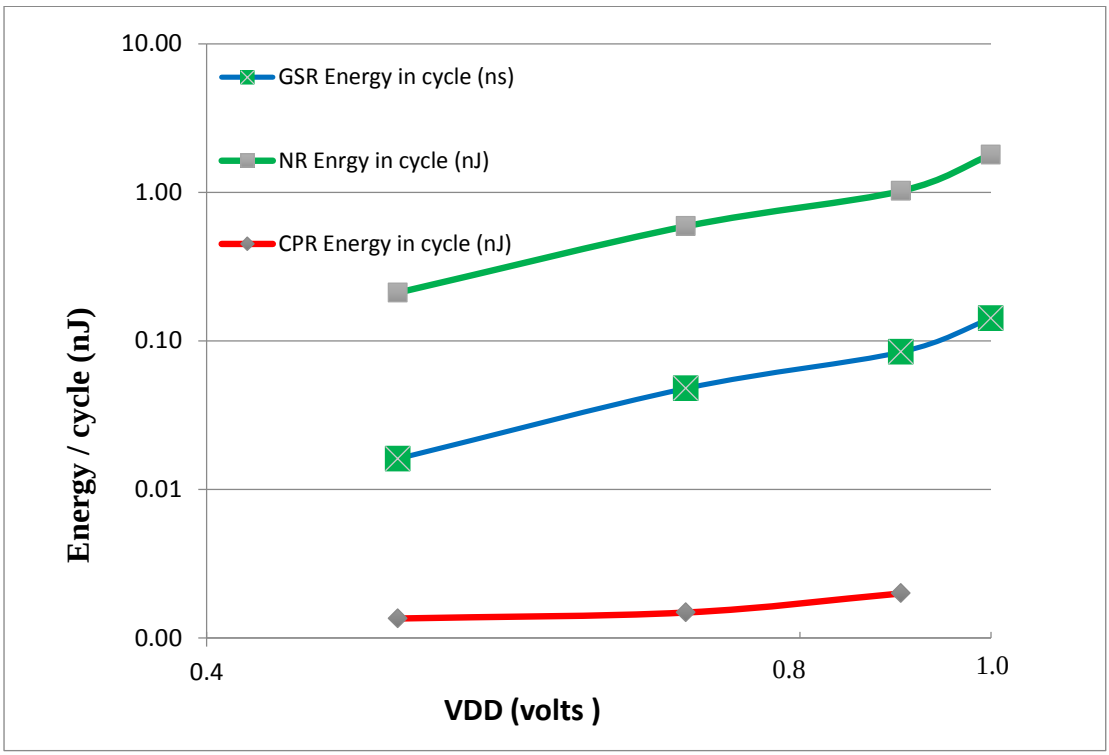


Figure 8.1 H-tree Energy per cycle with voltage scaling at 500MHz.

While NR needs no inductors, the resonance schemes need a characterized inductor L that sets $f_{RES} = 1/2\pi\sqrt{LC_L}$. For CPR, $f_{RES} = f_{CLK}$, so different inductor L_p values are needed to get minimum power at different clock rates. For a given L_p, the

frequency range of power savings is only an octave or so. This is a severe limitation in DVFS systems that aggressively scale down frequencies and supply voltages to the minimum needed at run-time. With large variations in load capacitances over PVT corners, even the best choice of L_p may not be optimal in actual operation without run-time tuning. Power savings in CPR over NR are not uniform, but frequency dependent, as shown in Table 1. For GSR and PSR, the resonance time T_{RES} need only be less than T_{CLK}. This inequality requirement enables the DVFS support by PSR and GSR. It also has the benefit of providing an extra degree of freedom for handling variations in C_L and L_S. The component Q_L (for frequencies before the onset of skin-effect [8]) is higher for PSR/GSR, than CPR, since resonance frequencies are higher.

8.1.2 Delays

NR gives the shortest propagation delay. The propagation delay of CPR driver is much larger than NR. This adversely affects skew and jitter due to the larger absolute variations and supply sensitivities. However the insertion delay for CPR can be comparable since the predriver requirement are much less. This can lead to lower jitter for CPR than other schemes. PSR and GSR resonate at much higher frequencies at the edges of the clock rather than the whole period like CPR, giving lesser propagation delay than CPR. The change of delay from supply variations is important for several reasons.

During run time the designer would like it to operate at supply voltage to meet the performance criterion so that power can be minimized. Having the information on which topology will give lowest power for a given delay requirements is helpful to determine which configuration to choose for GSR. Finally, it is to be noted that jitter is directly determined by sensitivity of the delay to supply variations (β) and it is desirable to operate in lower slopes of the curves in Figure 8.2. Increasing delay and

using parallelism allows for lower supply voltage and the corresponding power reductions from $C_L V_{DD}^2 f_{RES}$.

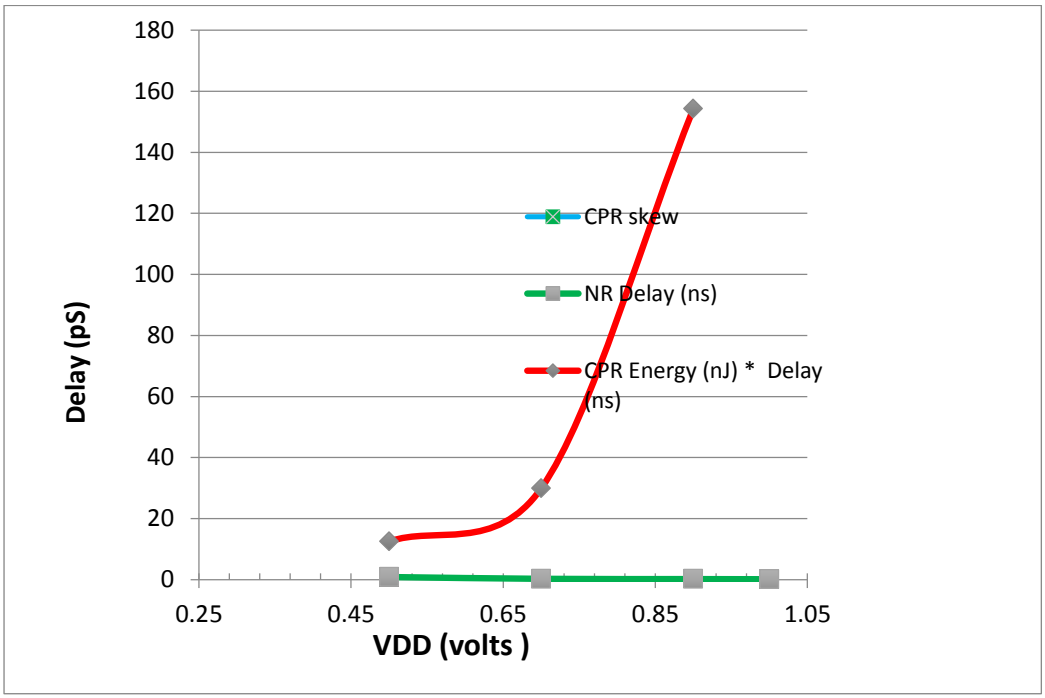


Figure 8.2 Delay variations with supply voltage.

8.1.3 Rise/Fall Times and Slew Rates

In resonant schemes, the rise/fall times depend on the resonance period T_{RES} ($T_{\text{rise/fall}}$= $0.29T_{RES}$). For CPR, this is nearly T_{CLK}, so the rise/fall times are long for lower frequencies, causing increased timing delays. This further leads to increase in power of the receiving gates due to short circuit currents. In contrast, since T_R in PSR and GSR is much smaller than minimuim T_{CLK}, the slew rates are fast, well controlled and fixed, resulting in low skew values. Again observing the change with supply voltage an optimum resgion of operation can be arrived at. Slew rates directly affect the clcok skew and more power is needed to achieve lower skew. While NR rise/fall times and slew rates depend on supply voltages slightly, the resonant schemes have these transition parameters fairly independent of supply voltage.

8.1.4 Skew and Jitter

Skew and jitter directly affect the timing budget. Figure 8.3 shows the skew variations over supply voltage. Once the minimum skew requirement is determined, the right topology and minimum supply voltage can be chosen.

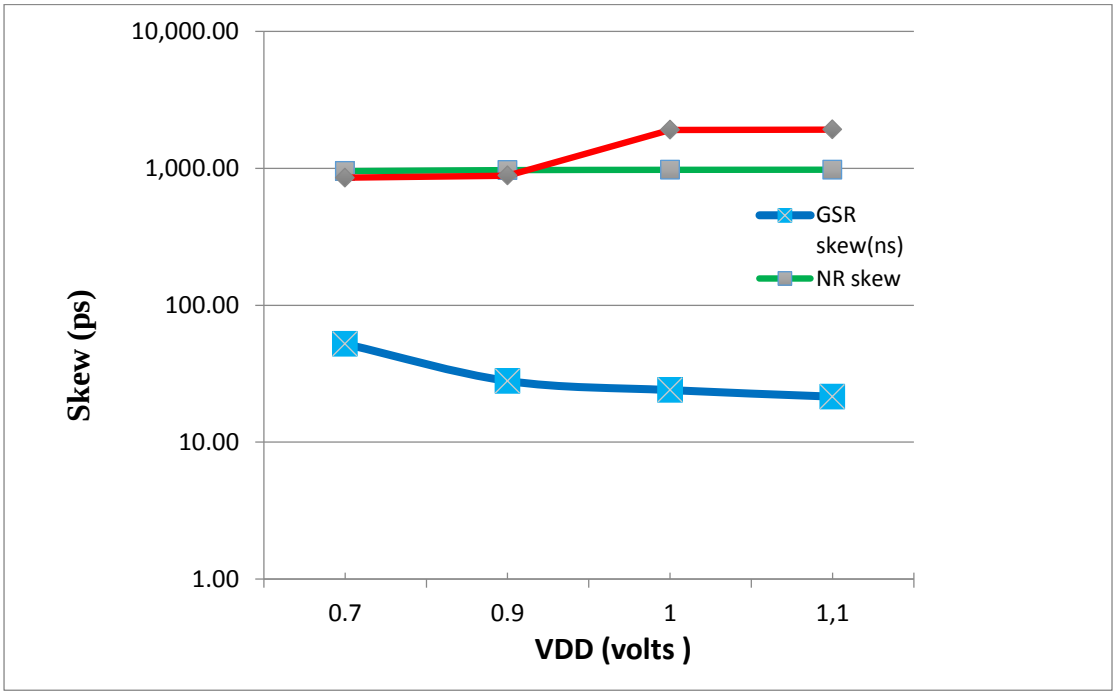


Figure 8.3 Skew variation with supply voltage.

8.1.5 Area of Driver

CPR drivers take less than 25% of the active area of an NR driver. PSR driver takes around the same active area as NR but needs extra metal area. GSR takes 25% more active area than NR and needs more metal area than PSR.

The inductor value for a given resonance frequency and capacitance is given by $1/4\pi^2 f_{RES}^2 C_L$. In GSR/PSR, giving some margin for pull up/down time, the resonance width ($T_R=1/f_R$) is usually set at about $1/5^{th}$ of nominal T_{CLK}, resulting in 5× larger value for resonance frequency than the clock **[**24]. The series inductor value is then smaller, given by $L_S=L_P(f_{CLK}/f_R)^2$. Both PSR and GSR need less metal area for inductors in the driver compared to CPR.

Inductor metal area for PSR and GSR can be on top of the driver active area and not encroach on other active areas. The inductor metal usage can sometimes affect critical performance due to routing blockages in the clock tree synthesis. PSR can also use bond wire inductors or off-chip inductors, especially for low frequency operation [24]. For comparison, NR needs 8 INVs to drive a load of 1pF with optimal delays; CPR takes less than 4 INVs; PSR takes 5 INVs and GSR 15.

8.1.6 Predriver Overhead

All the driver schemes shown need additional circuitry for input pulse stream generation. NR and GSR need non-overlapping pulses. CPR needs a minimum timing pulse width for a given driver size for proper operation **[**34]. Keeping the pulse widths minimum will minimize the static leakage in large driver devices. The predriver requirements are also important in determining total power and silicon area. When driving entire clock tree loads (>100pF), the matched capacitors in Fig. 4 can take excessive area. Making the inductors L_D and L_{PW} 10 times or more can scale the capacitance area down by 10×. Inductors' extra metal area is not considered as they can be stacked on top of the active area of the predriver.

The PSR predriver takes an equivalent of only 6 INVs compared to 16 for GSR. However, NR driver does need predrivers (nearly 5 INVs) to reduce delays in driving the large gate capacitance of clock drivers leading to tapered buffer sizes. In an NR H-tree clock distribution, the extra capacitance driven can be 50% of C_L for optimal delays, leading to 50% more power **[**20]. CPR buffer sizes are small compared to other schemes.

8.2 Energy-Delay (E-D) Tradeoff

Modern low power designs employ quantitative pareto analysis to arrive at best configuration and operating conditions. Combining the insertion delay and power

graphs into a combined metric of Energy-Delay product (or sometimes called speed/power metric) shown in Figure 8.4 allows for a holistic view of topology selection.

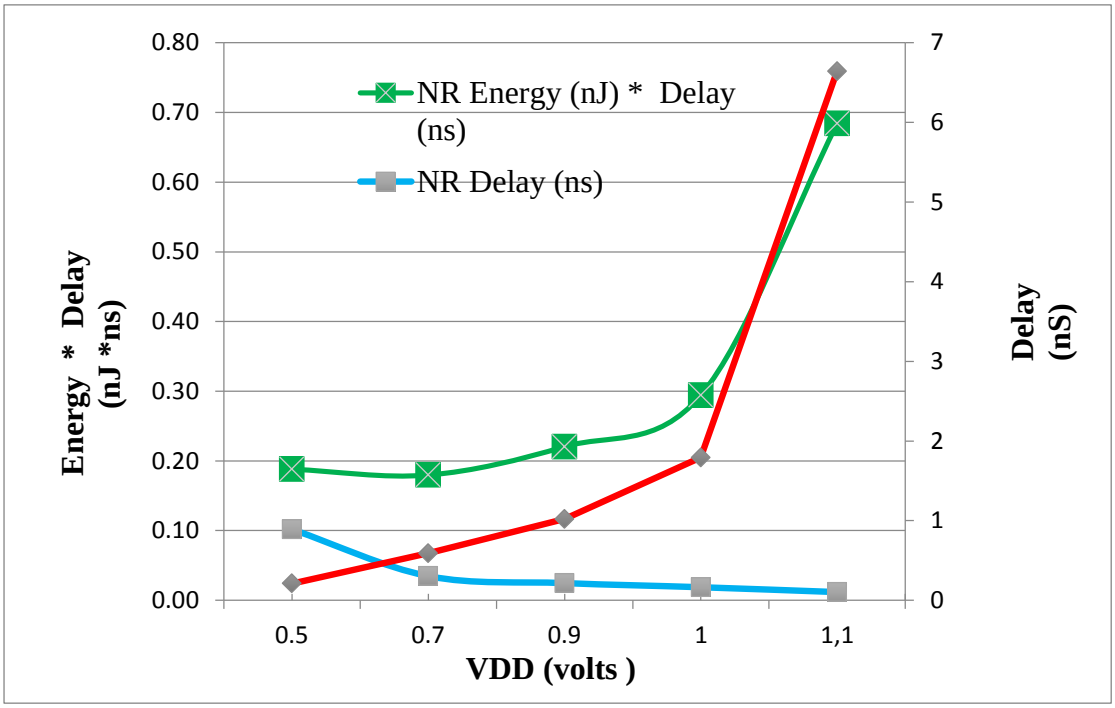


Figure 8.4 Deriving E-D product curve.

Figure 8.5 shows the Energy-Delay (E-D) product for NR, CPR and GSR to see the figure of merit of one over the other. CPR has the lowest (best) values since the insertion delays are the lowest due to little overhead in terms of predriver delay, although the driver itself is slower than other schemes. However the operating frequency is only valid over a small range of voltages over which frequencies around the resonance are supported. GSR is a good balance between NR and CPR.

By plotting energy vs. delay as in Figure 8.6 pareto analysis can be more effectively used. Area can be factored in the Pareto chart as well to do a comprehensive PPA analysis.

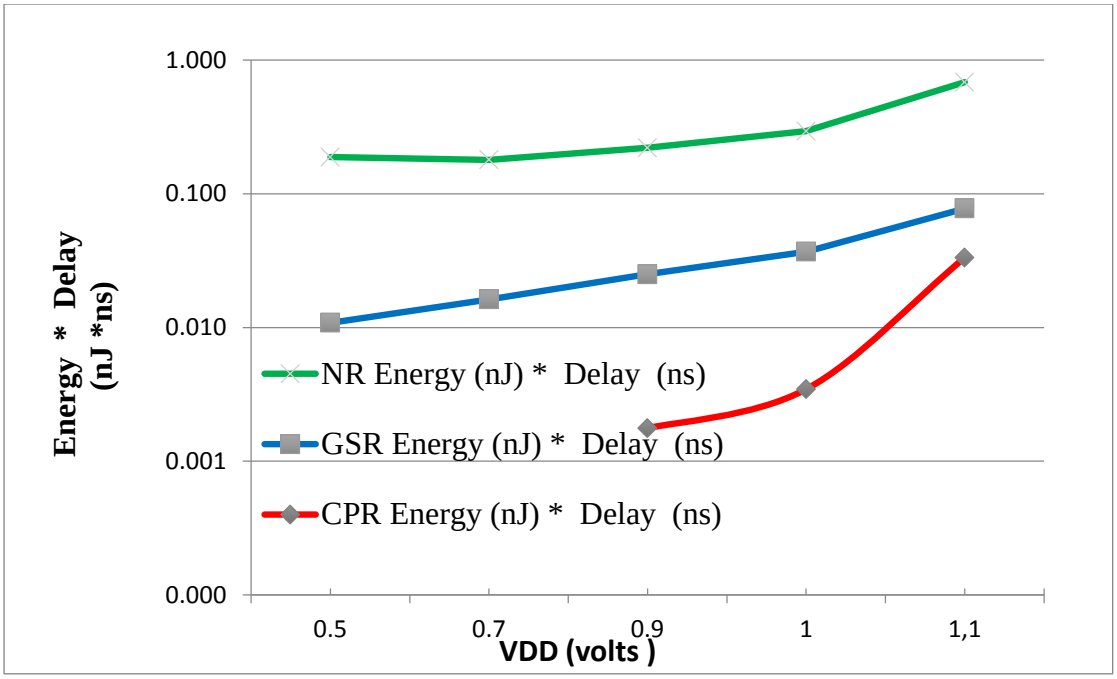


Figure 8.5 E-D Product for NR, CPR and GSR.

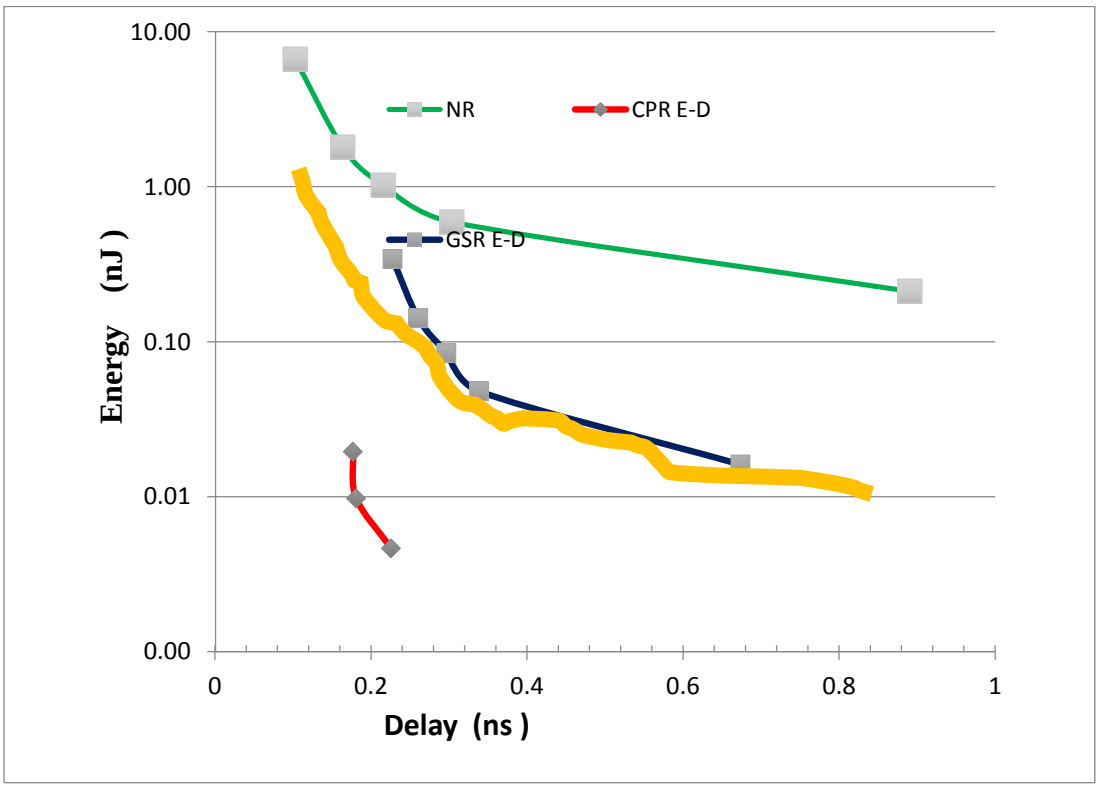


Figure 8.6 Pareto Graphs for Energy vs. Delay.

Energy-delay metric is usually improved with technology scaling with ‘More of Moore’. Figure 8.5 shows how it can be improved through the use of inductors which is basically a ‘More than Moore’ solution.

8.3 PPA Optimization

In what is commonly termed as PPA optimization, power, performance and area estimates, as shown in Table 1, are considered simultaneously. An optimal configuration (indicated by bold text) may be selected for the best performance or lowest power, at the frequency of operation. As an example, for low frequency operation, NR may be used since dynamic power is small and acceptable. PSR needs minimum driver and buffer sizes and is ideal for single frequency operation like in global clock distribution. For DVFS in regional clocks, PSR or GSR provides power savings at all clock rates. For DDR operation, PSR is the best, operating on both the edges of the clock. PSR has Q degradation compared to GSR. GSR, like NR, can drive standard gates without needing special buffers or latches and thus preferred over PSR for the current automatic synthesis tools. GSR may also be used in data path using dynamic logic for power savings as shown in Chapter 6.

8.4 Applications

Power consumed in post processing resonant clock waveforms may need to be considered for the given application. Due to the sinusoidal nature of the distributed clock signal, special flip-flops [21], [31], [32] are often needed to capture data correctly for CPR. PSR, on the other hand, may give additional savings in flip-flops with its pulsed outputs as described in next section. The pulsed output of PSR can drive simpler latches, instead of full master-slave flip-flops, saving more power and even area **[**23]. NR and GSR can drive standard gates.

CPR driver clock distribution is employed at global clock level as it takes least power near resonant frequency and least active area. Thus power is reduced with respect to NR, while the delay performance worsened and area decreased, showing a different tradeoff. PSR is well suited for double data rate (DDR) operation. PSR can operate with lower V_{LB} of $V_{DD}/4$ for low Q values. R_T needs to be kept low for series RLC to keep the inductor size small. This needs large size switches to keep Rr component small. Even for low Q of 2, more than 60% NR power can be saved using PSR. Accordingly, the tradeoff obtained for PSR is better performance than CPR, but with more power and active area.

Modern mobile and high performance designs are using increasing number of voltage domains and with regional clock trees and grids [18]. Thus, it is beneficial to improve and extend the globally-resonant clock drivers to locally-square non-resonant drivers in the CDN [14].

Resonant solutions, with characteristic sine wave signals, were initially applied to lower speed systems. Special flip-flops for ultra-low energy applications were designed to work with these low amplitude signals from global clock grids [21]. These custom cells need to be incorporated into standard cell libraries for synthesis. The power savings are further improved by dual edge-triggered (DET) operation wherein the clock speed itself can be halved and a lower supply voltage used. The tradeoff is in the extra transistors and area taken by DET master slave (MS) flip-flops.

9 SYSTEM LEVEL EXPERIMENTAL RESULTS

From a top down perspective power needs to be saved while meeting the timing requirements at system level for synchronous operation with a common clock. A poor clock distribution network can result in

- Limited speed due to setup timing violations
- Functional failures due to hold timing violations
- Large Power consumption due to excessive loads

Objective of CDN shown in Figure 9.1 top-down view is to distribute a clock signal to the sequential storage elements in a manner that, for every pair of flip-flops (*i*, *j)* through which there is a timing path, both the setup constraint and the hold constraints are satisfied as in equation (9.1) and (9.2) for timing closure.

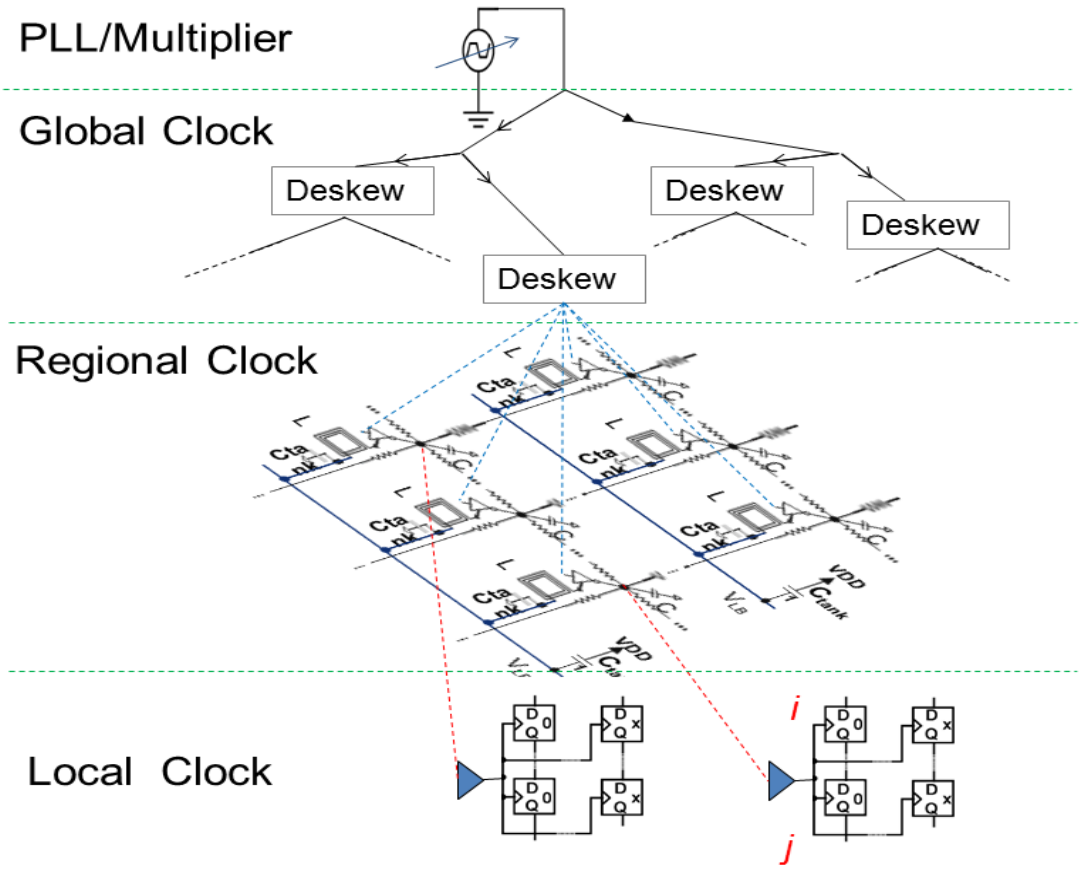


Figure 9.1 Typical Architecture of CDN.

$$D_{i,j} + t_{dCQ} \leq T_{CLK} - t_{skw\,i,j} - 2t_{jit} - t_{setup} \tag{9.1}$$

$$d_{i,j} + t_{cCQ} \geq t_{skw\,i,j} + 2t_{jit} + t_{hold} \tag{9.2}$$

In these equations, $d_{i,j}$ ($D_{i,j}$) is the minimum (maximum) data path delay between the sequential elements *i* and *j*, T_{CLK} is the clock period, t_{setup} (t_{hold}) is the setup (hold) time, t_{dCQ} (t_{cCQ}) is the clock to output delay (contamination delay) of a sequential element, t_{skw} is the skew and t_{jit} is the jitter. The *local skew* is $t_{skwi,j} = t_i - t_j$ from sequential element *i* to *j* where t_i and t_j are the delay of the clock signal to the sequential element clock pin, which is also called a clock sink. The maximum, minimum or average delay from the clock source to all sinks is also referred to as the insertion delay of the clock tree. Jitter is the maximum variation in clock arrival time at a sink [18].

As an example, Figure 9.2 shows the bottom-up view from flip-flops FF1 (*i*) and FF2 (*j*) with a common path delay from buffer predriver C1 and non-identical drivers C2 and C3 creating a skew. The data path delay $d_{i,j}$ ($D_{i,j}$) comes from D1 and D2. The $t_{skw\,i,j}$ will also include interconnect mismatch effects from *n1* and *n2* wires. The data path wires *n3* and *n4* contribute to $d_{i,j}$ ($D_{i,j}$) as well.

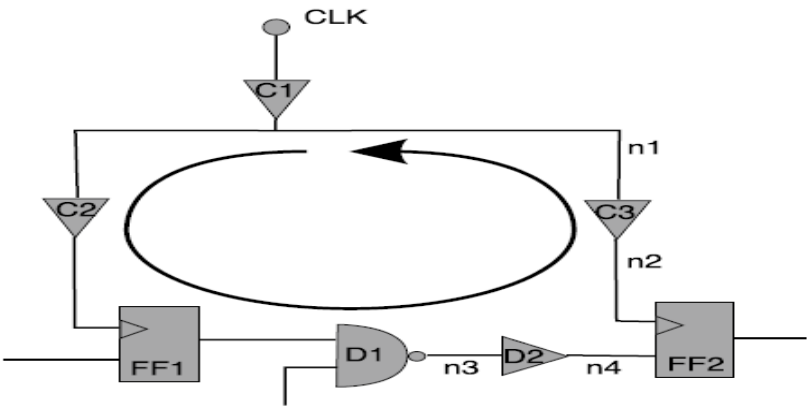


Figure 9.2 Bottom-up Timing Error Sources.

Figure 9.3 shows a standard benchmark recommended by IBM in ISPD2010 to evaluate skew in clock synthesis [32]. The target is a balanced H-tree but actual implementation mismatches the nominal length of 1.25mm by as much as 32%.

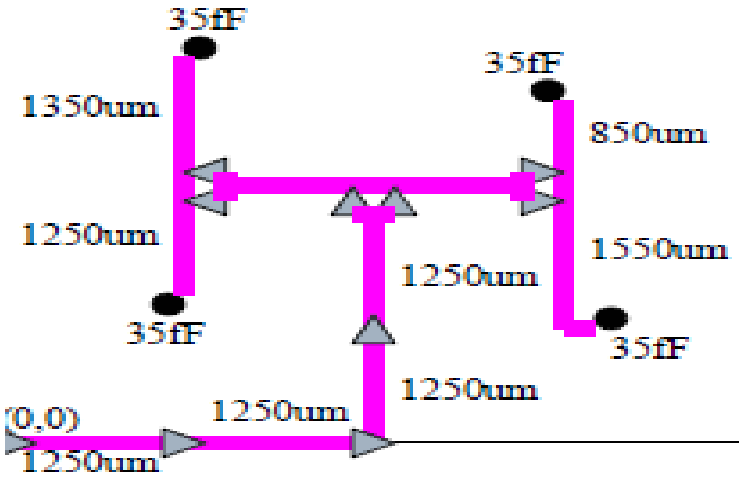


Figure 9.3 IBM ISPD2010 skew generation benchmark.

9.1 System Timing Closure

Combining Figure 9.2 and H-tree from Figure 9.3, one can visualize the C3 and C2 skew coming from two different branches of the tree driven by different buffers and interconnect that are of the same type but suffer from systematic and random on-chip variations (OCV).

Static Timing analysis (STA) evaluates the timing slack/margin of nodes and edges based on the difference of actual arrival times and required times. STA computes an upper bound on the delay of all paths from the primary inputs to the primary outputs, irrespective of the input signal combination. STA is a highly efficient method to characterize the timing performance of digital circuits, to determine the critical path, and to obtain accurate delay information. In Figure 9.2 example, STA predicts the earliest time when FF2 can be clocked, while ensuring that valid signals are being latched into all flip-flops and registers.

In Figure 9.2, for example, there are two choices to improve performance: speed up the clock to FF1 or slow down the clock to FF2. Without considering process variations, there are many options that have the same effect. For example, wire *n2* can be made wider so that it presents more loading to gate C3; gate C3 can be made smaller so that it has larger delay; or wire *n1* can be made narrower to increase its resistivity. These options are just for slowing down the branch to FF2 - similar options exist to speed up the branch to FF1. Deterministic STA optimization would do some combination of these moves to quickly converge to a solution.

Considering the process variations, however, some of these options are less attractive than others due to the correlation between the data-path delay from FF1 to FF2 and the clock tree skew between the clock nodes of FF1 and FF2. To make the design more robust, it is best that these two delays be correlated. If they are correlated, a process parameter will affect both the data-path delay and clock skew equally and, in turn, not impact performance. For example, if the data-path is gate delay dominated, one may wish to add extra delay in the clock tree by sizing C3. If, however, the data-path is metal interconnect dominated, one may wish to add delay in the clock tree by sizing a metal wire to improve the correlation.

If C1 is powerful enough to drive FF1 and FF2 only the interconnect mismatch would matter. If the end points of C2 and C3 were shorted (grid) again the mismatch would be minimized. Both these are utilized in Resonant clocking to control skew. A negative set-up time in the FFs gives an extra margin to the amount of harmful skew that can be tolerated.

Figure 9.4 shows the generalized model for statistical calculations. Typical timing analysis performs the setup and hold checks at the sampling flip-flop.

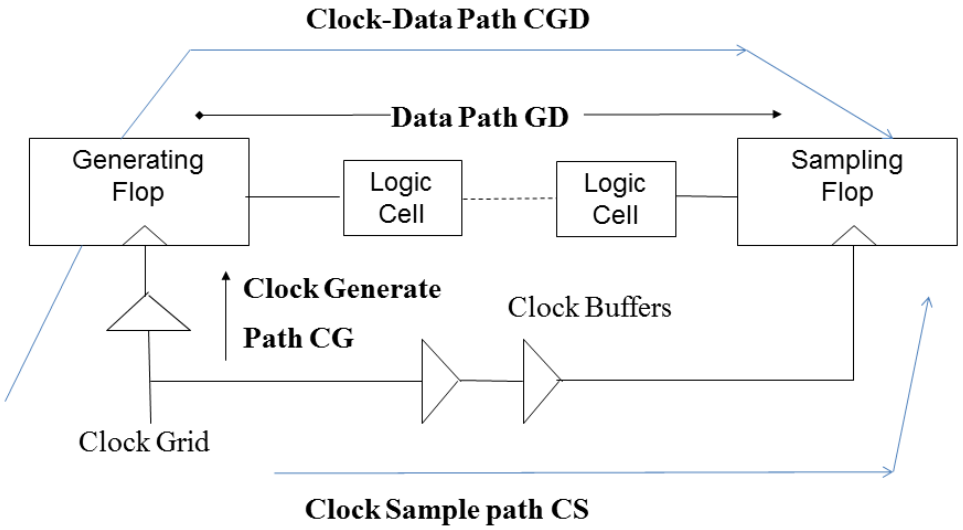


Figure 9.4 Generalized Statistical Timing Slack Calculations.

Simplifying the setup check as

$$setup: \quad t_{GDmax} + t_{setup} + t_{skw-max} \leq T_{CLK} \tag{9.3}$$

where $t_{GD\max}$ is the maximum possible delay of the path *GD*, t_{setup} the setup time of the receiving flop, T_{CLK} the desired cycle time, and $t_{skw-max}$is the estimated variation in skew for the slow process corner. A negative setup time in flip-flop or latch element will make it easier to meet setup constraints at highest clock speeds.

Similarly, the hold check is given by:

$$hold: \quad t_{GDmin} \geq t_{hold} + t_{skw-min} \tag{9.4}$$

where $t_{GD\min}$ is the minimum possible delay of the path *GD*, t_{hold} is the hold time of the receiving flop, and $t_{skw-min}$ is the skew for the fast process corner. Most timing analysis flows account for process variations in calculating these skews by applying a process variation penalty in addition to the nominal clock skew. This penalty can be derived from approximate first-order formulas based on the clock path delays or from skew tables which store pre-computed values of these skews.

One can actually compute the margin for the check according to statistical theory [46]. The basic failure mechanism for a setup check is that the time it takes for a signal to reach the receiving flip-flop via path *CGD* is greater than that of the sampling *CS* augmented by a cycle delay. This difference called the *margin* is a quantity that should be analyzed statistically. The reason for statistical analysis of margin is that differences are treated differently for statistical quantities than for deterministic quantities. The deterministic margin at the receiving flop is given by:

$$margin = t_{CS} + T_{CLK} - t_{GD} \tag{9.5}$$

Equation above is a valid equation for computing the mean of the margin. The variance of the margin according to statistical theory is given by:

$$\sigma^2_{margin} = \sigma^2_{delay,CS} + \sigma^2_{delay,CGD} - 2\,cov(t_{CS}, t_{CGD}) \tag{9.6}$$

Where $cov(t_{CS},t_{CGD})$ represents the covariance due to process variations in the respective path delays. It can be seen from above that the variation of data path delay adds to the overall variation which is in contrast to the subtraction of mean delay of path *CGD*. Moreover, a component of the statistical variation represented by the common variations of both paths, the covariance term in (9.6), can be used to improve the margin.

This recovery of margin because of the correlation in the systematic component of clock and data path delay variation allows for a less pessimistic (and more accurate) estimate of setup and hold margins thereby expanding the design window. For resonant clocking no active buffers are necessary so that delay matching to the data delay can be added to increase the covariance term and eliminate excessive guard-bands in the design. For purely random variations the covariance term is zero.

9.2 PSR vs. NR sub-system performance

The PSR naturally creates the controlled sharp falling edges. This can be seen from the PSR clock sampling in Figure 9.5. PSR can drive epTSPC meeting the requirements of robustness and controlled steep slew-rates. At system level, the predriver that generates pulses can be shared among multiple PSR drivers if the T_R requirements are homogenous among the drivers. Figure 9.5 shows the results for NR clocking with optimally sized tapered buffers driving the inputs of the 1024 flip-flops. Skew can be reduced as needed with wider interconnect lines, but at the expense of more power. The combined clocking and flip-flop operation is compared to demonstrate the equivalent throughput of PSR and NR schemes for same latency and skew. For PSR sized to drive the 1024 epTSPCs and interconnect with less than 10ps skew, a savings of 68% is seen when compared to NR with the same latency. This agrees with the theoretical calculations. For a t_{dCQ} of 48ps, epTSPC takes only 5.9fJ of energy per cycle driving a 5fF load at 1V supply, whereas the deTSPC needs 7.8fJ. This is a saving of > 26% in FFs while the overall saving is 45% for 1024 flops.

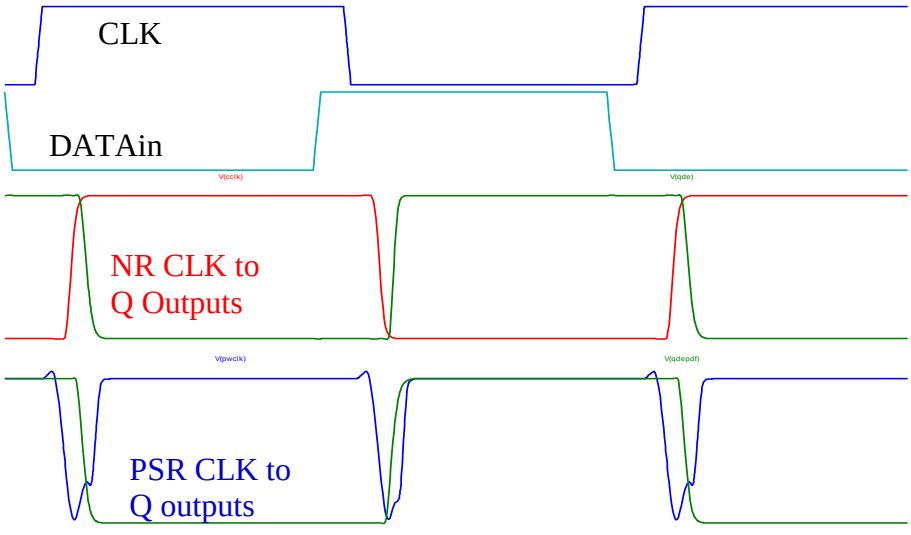


Figure 9.5 PSR vs. NR with same t_{dCQ}

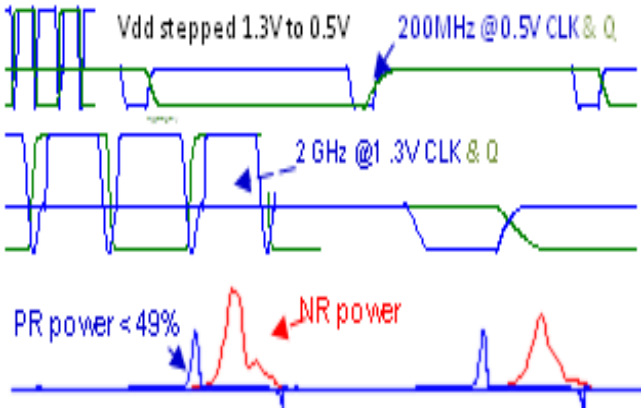


Figure 9.6 Power Savings over DVFS range.

The DVFS operation of PSR is verified over a decade of frequencies in the system as shown in the transient simulation of Figure 9.6. For DVFS operation, the clock frequency is scaled down to 200MHz supporting 400Mbps peak data rate at 0.5V. It is also scaled up to 2 GHz with 4Gbps at 1.3V. Figure 9.6 shows the functionality over the entire DVFS range and instantaneous NR power compared to PR power. Note that the horizontal/vertical scales are zoomed in for clarity for different signals with scaled voltages and frequencies. The PR dynamic power can be seen to be less than half of the NR power over the DVFS range.

The PSR-epTSPC and NR-deTSPC transistors are sized and designed using PTM 45nm devices. Test benches by IBM from ISPD2010 clock synthesis are used, which include interconnect parasitics [32]. A fan out of four (FO4) loading (5fF) is used and the supply voltage varied from 1.3V to 0.5V. Extensive simulations in SPICE with PTM 45nM devices verify operation of the PSR-epTSPC for power savings and skew control. The complete leaf cell implementation in 45nm of the 1024 flops clocked by PR through an H-tree network was used for post-layout simulations. Figure 9.7 shows the worst case of combined simulations of pulse generator and latches. Top of Figure 9.7 shows the early clock and late data (150ps skew) stress test condition for worst case timing. Simulations are for 30% Monte Carlo variations and

temperature sweep from 25 ^{O}C to 125^{O}C. Comparing the data capture operation at both the rising and falling edges, NR with DET FF fails to capture data in some corners when there is no set-up time before clock edge. PR with epTSPC captures the data correctly in all cases, even with negative setup time. This can be used as an advantage for clock de-skewing purposes. This reduces the width of interconnect lines needed to meet a given skew specification resulting in lower load capacitance and power. The hold time for epTSPC is well defined by the width of the resonance pulse and the clock to Q propagation (t_{dcQ}) is 4 inverter delays. This allows for predictable operation and timing closures.

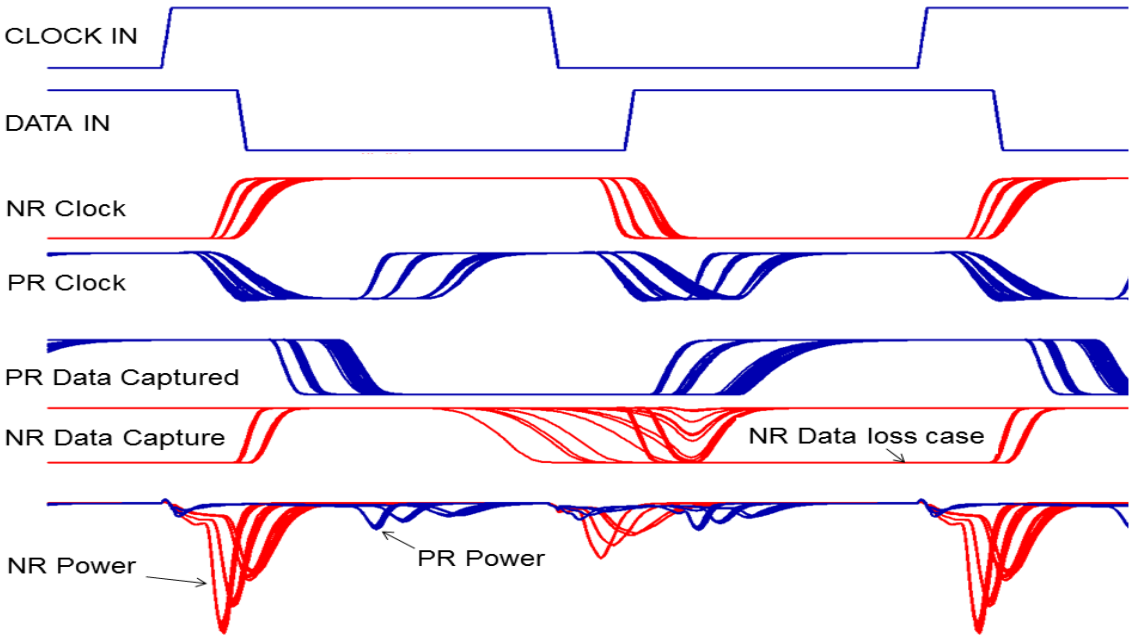


Figure 9.7 PVT and MC skew simulations comparing PSR and NR H-Trees.

Power and energy curves are derived as shown in Figure 9.8. The top curve shows the percentage power savings for PSR driver (PRD) over NR for clocking. The energy- delay product on right vertical axis shows 300fJ.ps at 1V and 1GHz compares well with metrics reported [31].

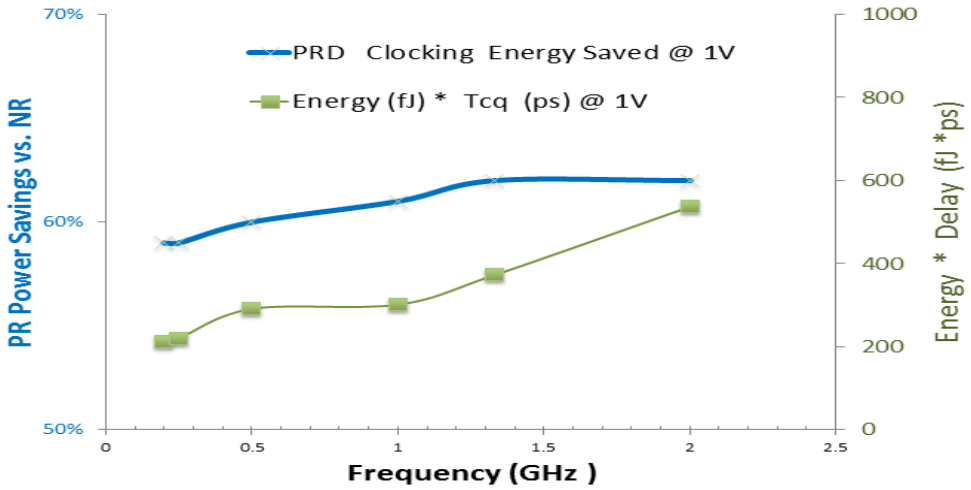


Figure 9.8 Power Savings and Energy.

Figure 9.9 compares the data capture edges with the clock leading data at both the rising and falling for repeated Monte Carlo runs. NR with deTSPC fails to capture data with no set-up time. PR with epTSPC captures the data correctly even with the negative setup time. This can be used advantageously for clock de-skewing purposes. The hold time for epTSPC is well defined by the width of the resonance pulse and the clock to Q propagation is 4 inverter delays. Thus, the clock to Q propagation can be kept larger than hold time to minimize hold time violations for timing closures.

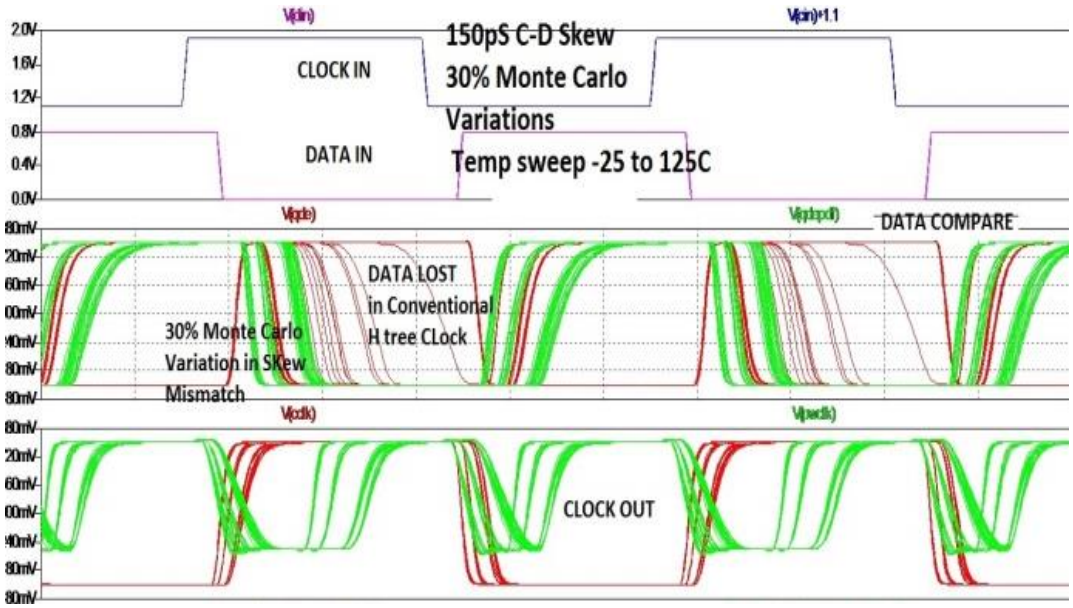


Figure 9.9 PVT and MC skew simulations showing PSR advantage.

A robust wide-frequency clock driver based on pulsed resonance driver (PRD) topology that consumes 60% less power than a conventional driver horn is demonstrated for local buffering. The PRD can work with standard latches (epTSPC) in DET applications taking 40% less area and power for 1024 flops compared to the current schemes. Negative setup time of epTSPCs give extra margin for skew management.

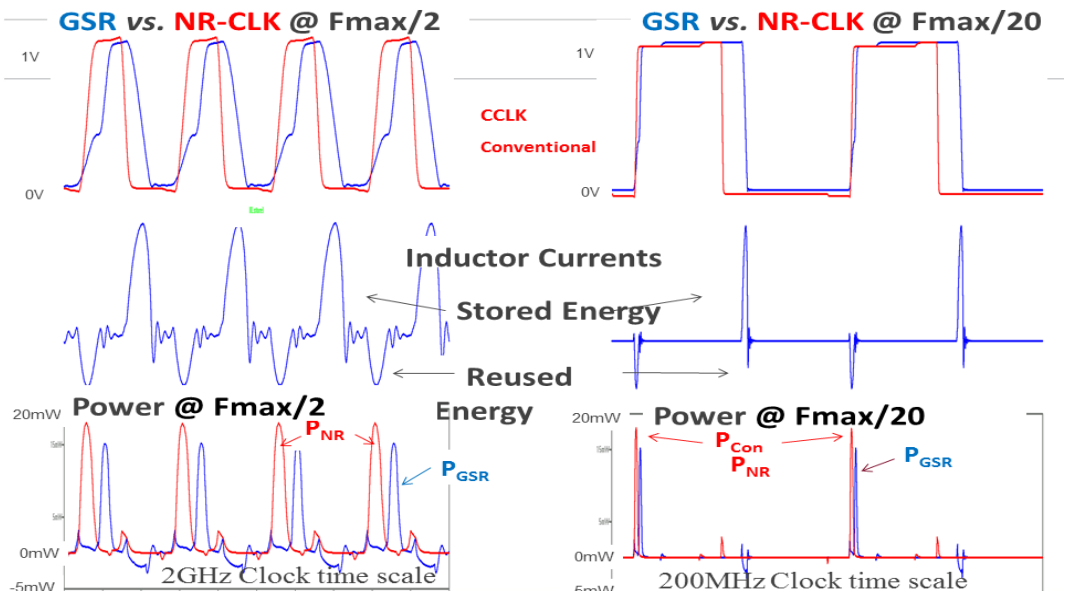
PRD itself can drive lower skew wider interconnect lines with less power. Small inductor values sufficient for pulsed resonance make this solution an attractive option for multiple voltage and multiple frequency domain regional clocks. As with CPR, issues can be progressively resolved in silicon for PSR. Though silicon measurements are not available at this time, the simulation results match well with the theory developed and corroborate well with previous silicon results re-simulated under same test benches of the bench marks.

9.3 GSR vs. NR sub system Performance

Dynamic power evaluation on 45nm IBM compatible process from ISPD2010 bench marks is chosen as a test case. A CDN, scaled for a 45nm, is simulated for more than a frequency decade below the maximum operating frequency (F_{max}) of 4GHz. Power savings over a 10× frequency range of the GSR configured as a wide frequency resonant driver are compared to those of a NR driver in Figure 9.10. In (a) 2GHz GSR operation with power savings over NR is shown while in (b) 200MHz GSR operation with power savings over NR is shown.

For a direct comparison, the NR and GSR are sized to drive a 1pF load. Though power is needed for the pre-drivers of both GSR and NR, they in turn eliminate short circuit currents that would have consumed larger power. The average

energy per cycle of P_{GSR} (<1.4mW) in a fixed interval for GSR is less than that of P_{NR} (>2.5mW) of NR. This can be seen from comparing the total area under the P_{NRD} and smaller P_{GSR} curves in the bottom row of Figure 9.10. GSR does need current from V_{LB} bias supply, but puts it back during discharge cycle, as seen in the negative excursions. GSR saves power for both the frequencies of 2GHz in Figure 9.10 (a) and 200MHz in Figure 9.10 (b).



(a) 2GHz operation (b) 200MHz operation

Figure 9.10 Power Savings over 10× clocking frequency range in 45nm.

The functionality and robustness of the new GSR driver and pre-driver circuitry is also verified by 22nm SPICE simulations across 30% variation in *LC* component values and transistor model parameters. The input drive of the resonant schemes can take power when large loads are being driven. The skew requirement between clock sinks often sets the drive strengths needed. Figure 9.11 shows the launched waveforms and the skew in arrival at the flip-flop clocking nodes for the three driver schemes.

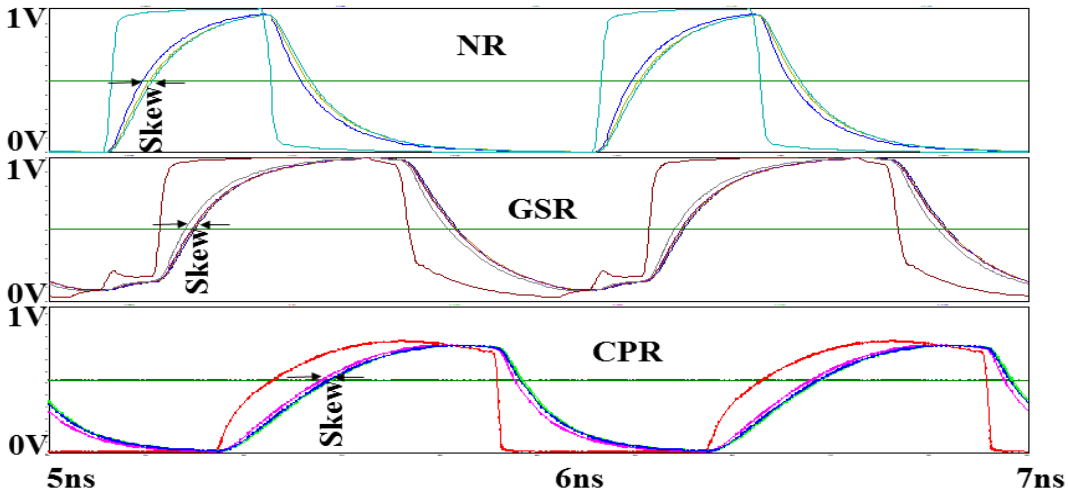


Figure 9.11 Variations in the delay contributing to clock skew.

Skew is minimized for NR, GSR and CPR with wide interconnects. A nominal skew of less than ±20ps is targeted for all to compare power required in 22nm.The skew from unequal loads are made to be smaller for NR, CPR and PSR by proper sizing and wire widths.

9.4 GSR, PSR, CPR and NR Comparative Analysis

In order to verify the tradeoff presented, the various clock drivers are tested under identical IC implementation parasitics from a symmetric H-tree benchmark [23], [32]. The resonance inductance values are derived from a standard metal spiral inductorof 0.5nH with $r_S < 10\Omega$ with a $Q_L > 3$ at 5GHz [8], [11]. The clock tree global interconnect is distributed on a metal layer with wires that typically have 0.1Ω/μm resistance and 0.2fF/μm capacitance. Clock distribution is done using 6 segments of 1.25mm each with 8 wires in parallel to reduce the nominal interconnect resistance to less than 2Ω. A ±30% random variation in length is considered for determining the clock skew. By keeping effective series resistance $R_T < 0.2\Omega$, a tank $Q > 1$ is obtained, which is sufficient for successful GSR operation. The effect of finite component Q_C (> 30) of the load capacitance is also factored in the simulations in terms of ESR.

For a 1V nominal operation, driving a distributed load totaling 160pF, Figure 9.12 compares NR, CPR and GSR power consumptions calculated across frequencies using SPICE simulations. GSR has L_S = 6pH and r_S < 0.1Ω @ 5GHz and CPR L_P = 160pH and r_S < 0.3Ω @ f_{RES} = 1GHz for V_{DD}=1V. Dotted lines show theoretical calculations. CPR is optimal at its resonance frequency f_{RES} and is not operated below 0.8f_{RES}. Inductor sizes are constant for CPR and GSR during the frequency sweep. The predriver power is included in Figure 9.12 in order to see a direct comparison between driver solution use-cases. Multiple unit inductors of 0.5nH are distributed in parallel along the tree to get the low 6pH value required to resonate at 5GHz. In Figure 9.12, GSR trend follows (3.10) and the NR and CPR track the theoretical equations for P_D from Table 1. NR takes the highest power (P_D), GSR less, and CPR takes the least.

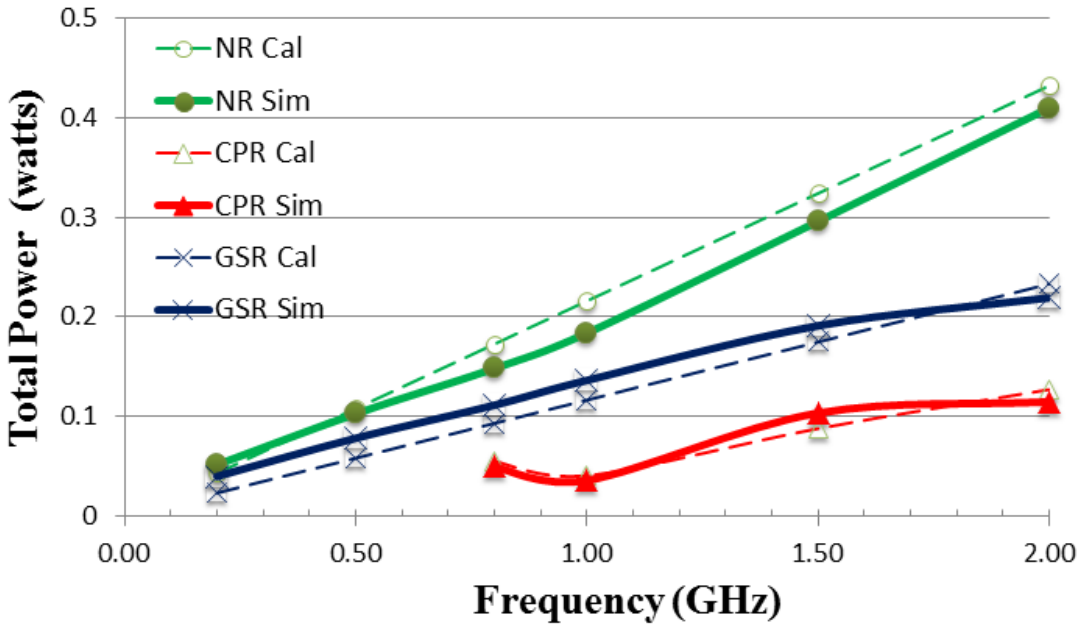


Figure 9.12 Power consumption versus frequency for NR, GSR and CPR.

The global interconnect lines reduce the output swing at higher frequencies due to *RC* delays as seen in Figure 9.11. This can result in lower power than

calculated. NR predrivers can improve the attenuated swing and minimize delays using tapered buffers, but at the expense of 50% more power.

Table 1 shows GSR predriver power overhead (P_P) of about $0.2\ C_L V_{DD}^2 f_{CLK}$. GSR driver takes about 50% of NR driver power of $C_L V_{DD}^2 f_{CLK}$. At 2GHz, as seen in Figure 9.12, total GSR simulated power (P_D+P_P) is about 57% of NR power, compared to 47% from Table 1 calculations. While the lumped model analysis is only accurate to 20%, it shows the comparative benefits of one topology over another.

The actual power values from simulations are also be different due to voltage dependent non-linear capacitances not accounted for in the theory. Short circuit currents in the NR predriver tapered buffers also cause deviation from the theory. It can be seen from Figure 9.12 that, as the propagation delays and rise/fall times get larger across topologies, less power is consumed by GSR and CPR, compared to NR, at higher frequencies. This is similar to the principle of adiabatic reversible logic, where slower transition times can give power savings [9].

Receiving local buffers will have varying logic thresholds that will cause appreciable skew for large slew rates. These thresholds will also vary due to dynamic supply variations causing jitter. For minimum skew, it is preferred to drive NR without distributed predrivers. Similarly, GSR and CPR with all inductors at source give minimum skew. However, due to *Q* degradation, this will consume more power than inductors distributed at sink points.

Figure 9.13 shows skews extracted from simulations over the DVFS frequency range for 160pF H-tree for topologies at 1V operation. Skew is the highest for CPR which has the largest power savings. NR has 10ps more skew than GSR.

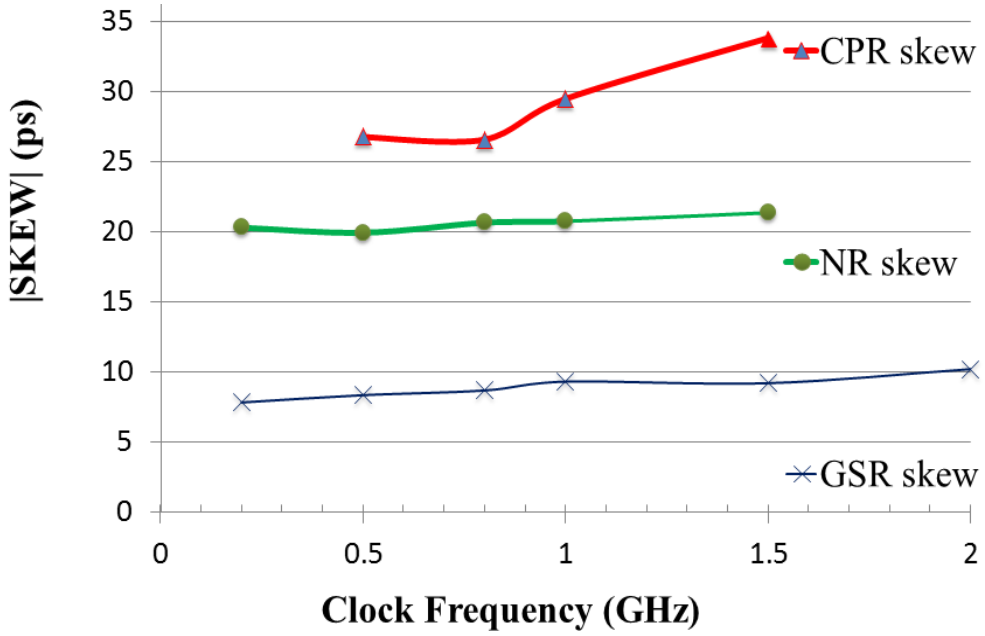


Figure 9.13 Simulated skews of H-tree across operating frequencies.

This is the true clock performance for a given power that needs to be considered. The GSR can give the lowest skew all the way to 2GHz, using the well-controlled falling edge as the trigger. CPR shows the highest skew and, like NR, cannot achieve functional swing at 2GHz.

With wider interconnects, target skew and functionality can be met in CPR, and NR as well, but at the expense of significant increase in the load capacitance and power [3], [18]. This again illustrates the fundamental trade-off between energy and delay, as one has to be increased to decrease the other. GSR gives low power performance below the resonance frequency f_R. However, with run-time reconfiguration to CPR, using the same inductor, its operation can be extended to f_R.

Figure 1.2 is the basis for a high performance CDN Mesh/Grid with DVFS operation from 2GHz @ 1V to 500MHz @ 0.5V. It saves more than 25% dynamic power on 45nm process from ISPD2010 bench marks. GSR based solutions have Run-time Digital Tuning capability for power and skew optimizations by varying resonance pulse width T_R. Resonance is achieved with smaller inductors occupying only the top

metal area [23]. The inductors are placed in the bottom rail of resonant drivers. A fairly large clock mesh capacitance of 1nF is targeted. Figure 9.14 shows the power savings for both 1V and 0.5V operation for GSR implementation across a wide frequency range, shown in log scale.

Figure 9.14 also compares simulated power savings of GSR with various conventional continuous resonant driver (CPR) solutions. Re-simulations of previously reported CPR solutions for global clocks in 90nm [14] and 32nm [11] are done under identical test conditions. The peak frequencies of CPR can be larger than f_R of GSR even for a slower process like the 90nm shown. The 32nm CPR curve shows narrow band of operation but good power savings at the resonant frequency, as verified by silicon measurements [11].

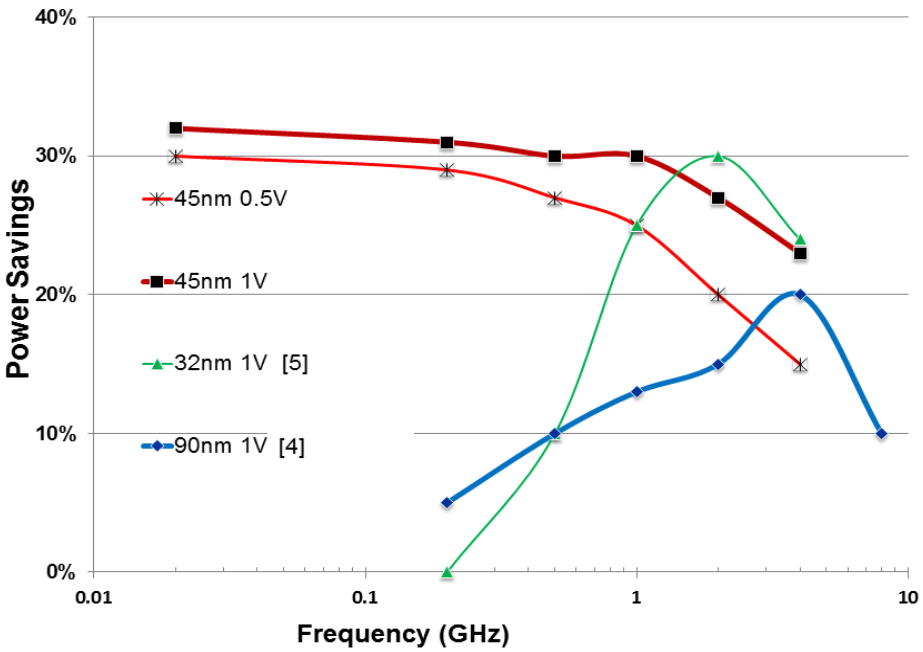


Figure 9.14 GSR Power Savings compared to NR.

As seen, GSR has an order of magnitude frequency range advantage over CPRs in maintaining power savings. The design has been verified over 90nm, 45nm and 22nm nodes and is thus seen to be readily portable across process technologies.

Table 2 summarizes the advantages and constraints involved in various driver choices and system level trade-offs. These have been simulated and validated in this chapter. As shown in Table 2, each scheme has its own unique advantages depending on performance needs and power. But all the schemes can be dynamically reconfigured from GSR. Only NR and GSR can drive standard cells with their outputs.

Table 2 Advantages and Constraints

	Non Resonance (NR)	**Cont. Parallel Resonance (CPR)**	**Pulsed Series Resonance (PSR)**	**Generalized Series Resonance (GSR)**
Flip-Flop Needs	Standard Library Cells	Extra Local Buffers or Sense Amp Flip-Flops [31]	Lower power with TSPC latches	Standard Library Cells
Support Circuits	Repeaters for less delay but more skew	$V_{DD}/2$ bias supply or $10\times C_L$ Decoupling Caps	V_{LB} bias Pulse Generator	VLB bias Pulse Generator Voltage doubler
DVFS	Yes	No	Yes	Yes
Auto Place & Rout	Yes	In development	To be developed	To be developed
Other Constraints	Unbuffered Tree Drive needs large power.	Larger power than NR at low frequencies. Large power for low skews. Large inductor sizes. Timing Closure issues.	Pulsed output not 50% duty cycle.	More circuitry and input waveforms
Key Advantages	Standard Flow	Unbufferred Driver. Lowest area. Less jitter causing Harmonics	Controlled edges to drive low power latches	Rail to rail output. Lower skew for single un-buffered driver.

This generalized series resonance (GSR) technique achieves 50% less power dissipation than NR drivers, while reducing the skew by 50% for meeting timing requirements. This series resonance schemes supports DVFS operation and has several advantages over parallel resonant drivers (CPR) as shown in Table 2.

10 DESIGN METHODOLOGY AND FLOW

The standard design flow shown in Figure 10.1 needs to be enhanced to include the resonant clocking with the best choice of configuration, inductors, driver sizes and placement. As a baseline NR solution is computed first as supported by most clock tree synthesis (CTS) tools.

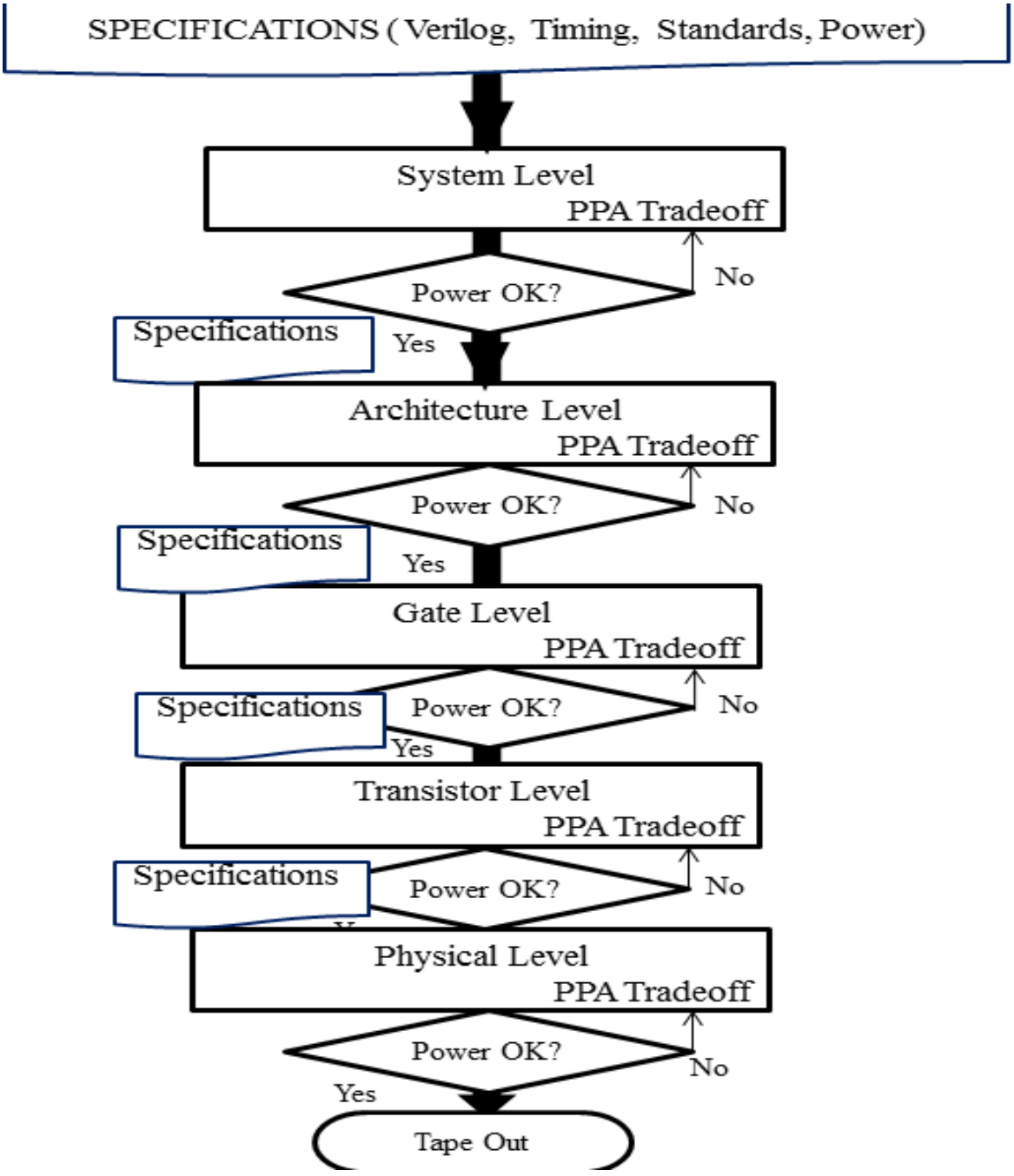


Figure 10.1 Standard IC Design Top Down Flow.

In a typical design flow, each design stage specifies certain characteristics that have to be implemented at the next level. Timing closure needs to be obtained in the final Physical Level stage.

Power consumption can be reduced by the designer at every stage by trading off area and/or performance (PPA). This requires that the power be estimated accurately at each stage. The equations derived in this thesis enable that. The accuracy of estimation needs to increase as the design progresses down the stages.

Resonant topologies will involve gate level, transistor level and physical level design stages. The split driver topology will need routing of symmetrical lines, in parallel, to the sink points of local buffers. This is used as the baseline solution to fall back on if the resonant schemes do not give appreciable power savings for the given skew and area limitations.

The algorithm for CPR inductor design and placement is in Appendix D: Design Synthesis. If DVFS of 5× or more is desired, PSR is the ideal solution along with custom latches, especially if DDR is used. The overall algorithm for PSR is similar to above as shown below:

Algorithm P: Overall PSR Synthesis Methodology
Input: Near-zero skew routed tree with LCB at root & **Grid** nodes from C1, f_{CLK} ;& DVFS, Lmin-max, MA-max (inductor metal area);
Skew t_{skw} constraint
Output: Inductor sizes and buffer locations

1. *taperWires()*
2. *while |Vswing| < V(minSwing) do*
3. *Vbest ← 0,*
4. *sizeLCTanks()*
5. *sizeDriver()*
6. *Run SPICE*
7. *if |Vswing| > Vbest then*
8. *Vbest ← |Vswing|*
9. *end if*
10. *end for*
11. *sizeLCTanks()*
12. *end while*

13. Place tank at n.
14. Run Spice
15. if min V(sinks) > V(minSwing) then
16. maxSwingNode = n
17. minSwing =min V(sinks)
18. end if
19. Remove tank from n.
20. end for
21. Place tank at maxSwingNode

If custom latches are not feasible and DDR is not employed, GSR can be chosen and the algorithm follows PSR algorithm. These are shown in the flow chart of Figure 10.2 as integrated into the main IC design flow. This can be incorporated into Automatic Place and Rout (APR) software as a low power design flow.

The following appendices contain more information of the flow and design synthesis.

Appendix C: Spread Sheet for Design shows a spread sheet that determines the basic feasibility for the given specifications.

Appendix D: Design Synthesis algorithms.

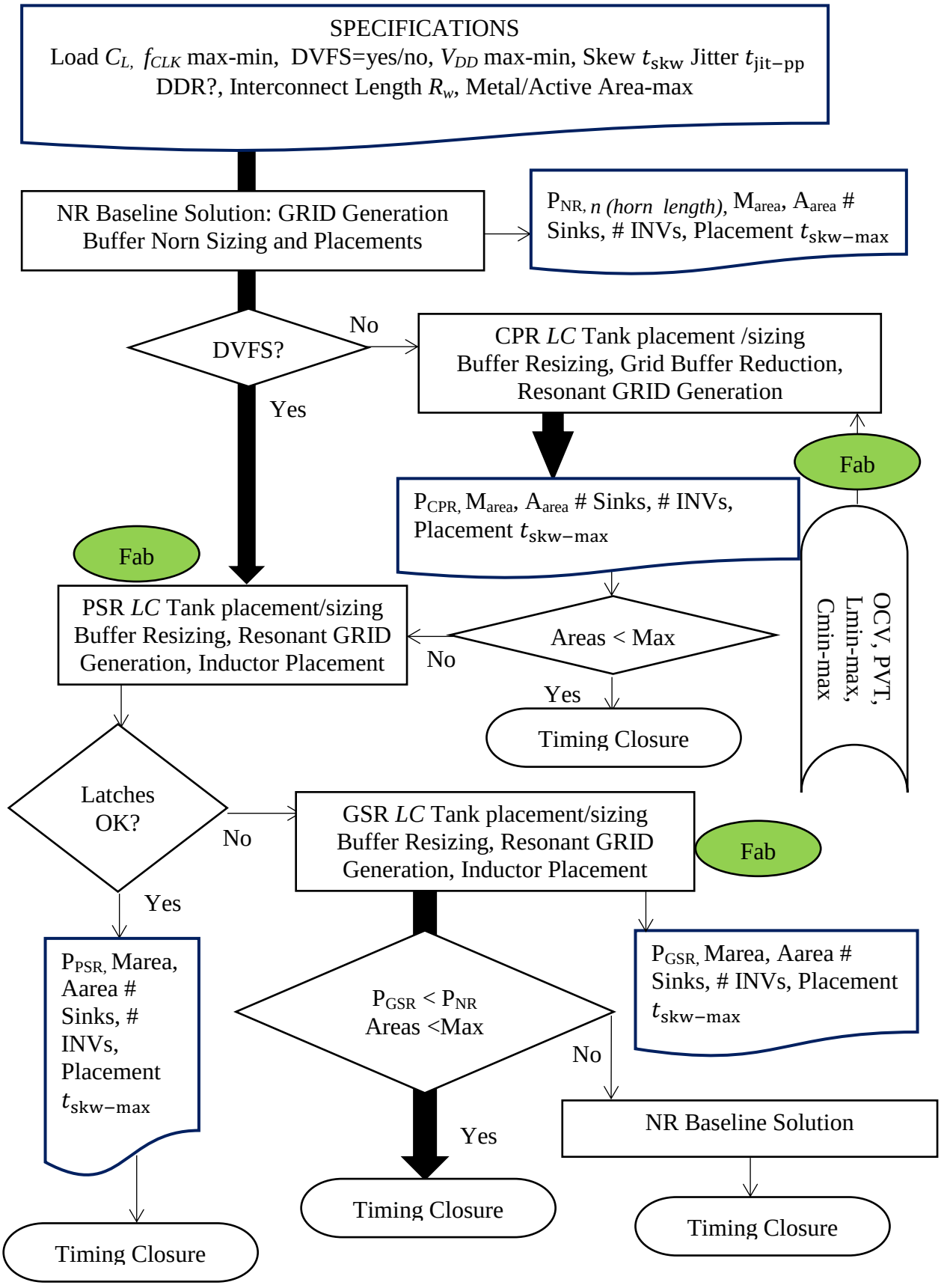


Figure 10.2 Design Flow for Energy Recycling Resonant Solutions.

11 CONCLUSIONS

As stated in the motivation section 1.1 of this dissertation, resonant solutions that inherently work over the entire DVFS range have been demonstrated in terms of the PSR and the GSR. The timing performance of PSR in terms of setup time, skew and jitter are superior to other solutions. PSR saves area as well using the TSPC designs shown. GSR improves skew and jitter but at the expense of area used. GSR can be used with standard library cells and reconfigured dynamically to other resonant and non-resonant schemes.

11.1 Summary

In summary, the GSR can be considered equivalent of a general purpose operation amplifier for clock distribution applications. The GSR driver gives rail-to-rail outputs that can directly interface to standard cell library flip flops and logic, and also allows clock gating. It has digitally controlled pulse width tuning for inductor variations, fast slew rates and lowest skew for a given power consumption. GSR can be reconfigured to give other schemes like CPR, PSR and NR. The only downside, if any, is the increase in area for GSR and metal inductors used. In this era of 'dark silicon' this is an acceptable compromise. In fact, increased area can reduce power density.

All the important circuitry for realization of the drivers was described to enable the drivers' deployment. Design equations for delay and power based on theoretical analysis have been derived and listed in Table 1. These are verified to be accurate with simulations on 90nm, 45nm and 22nm process nodes. All the sources of power consumption and delays in implementing resonant and non-resonant schemes are accounted for and compared. The performance, power and area (PPA) tradeoff for different schemes can be directly seen from the comparison charts to select an optimal

solution for the given application. To the author's knowledge such a comprehensive comparative analysis has not been attempted so far.

Additional receiver circuitry is needed by the resonant clock waveforms in CPR and PSR. CPR for example, needs specialized drivers or flip-flops that can handle non-square clock waveforms. The pre-driver of the series resonant schemes can take more power when large loads are driven by the driver. PSR actually takes less power than NR across the DVFS range, both for resonant clocking and flip-flops. The skew reductions are achieved without needing to increase the interconnect widths thanks to the negative set-up times.

Validation of PSR and GSR area also shown on a 45nm with layout plans to illustrate the scalability of the design. A comprehensive top down solution for applying resonance in clock and data timing is discussed. As the resonant inductor is used only during the rise and fall times, smaller values of inductors are sufficient and a decade of operating frequency range is possible. This allows for seamless DVFS operation that runs at lower voltages and frequencies to dynamically scale power consumption in high performance processors. Smaller inductor values of series resonance schemes make them an attractive option for multi-voltage and multi-frequency local clocking solutions. With sufficient unused top metal layers area, the inductors can be realized with little active area penalty.

A dynamic logic circuit RDL that uses GSR principle is also shown. Other dynamic logic circuits can also be combined with GSR for power reductions at functional level. This topology can also be used in driving the large capacitance that results in the word-lines and bit-lines of memory arrays. Inductors can also be shared between multiple drivers.

This work does not necessitate the use of high-*Q* custom inductors that need more active area or specialty processes. With reasonable tank *Q* values (>3), practically realizable on-chip, GSR solutions presented here can recycle more than 50% of driver energy over the entire DVFS range, reducing clocking power at system level by 40% on average. This LC resonant clock driver is shown to save power on a 22nm process node and has 50% less skew than a non-resonant driver at 2GHz. It can operate down to 0.2GHz to support other energy savings techniques like DVFS. There is less than 25% area penalty on GSR drivers.

Use of PSR and TSPC latches can further reduce the system power by another 25%. As an example, GSR can be configured for the simpler pulse series resonance (PSR) operation to enable further power saving for double data rate (DDR) applications, by using de-skewing latches instead of flip-flop banks. A PSR based subsystem for 40% savings in clocking power with 40% driver active area reduction was demonstrated. Simulations using 45nm IBM/PTM device and interconnect technology models, clocking 1024 flip-flops show the reductions, compared to non-resonant clocking. DVFS range from 2GHz/1.3V to 200MHz/0.5V is obtained. The PSR frequency is set >3× the clock rate, needing only $1/10^{th}$ the inductance of prior-art *LC* resonance schemes.

11.2 Conclusion

The stated goal of this thesis was to arrive at energy recovering resonant solutions that inherently operate over wide frequencies and give better performance in terms of lower skew and jitter for timing closure**. The dissertation has shown how to achieve that using GSR, with detailed theory and implementation.**

A typical processor bench mark has 25% allocation for clocking and 20% for flip-flops [1], [11]. With the PSR-TSPC solution demonstrated it is possible to save

40% of power amounting to 18% of system power. This amounts a decrease in temperature rise above ambient by 18%. Failures are accelerated with temperature and this can amount to a 10% decrease in failure rate. It also allows for choice of more economical packaging and 10% lesser cooling costs for the end customer. For the IC vendor, yield is improved due to decrease in area as well as the improved margins in timing performance. A 40% decrease in clocking and flip-flops area gives effective die size savings of more than 10%. Die cost decreases proportional to 4^{th} power of die area giving a cost savings of 35% [27]. Adding increased performance margin in timing can take this to 40% savings in die-costs, including testing, when compared to NR based DDR designs. So cost savings are realized along the whole chain from IC manufacturer to the end equipment user.

Standard DSM CMOS implementation of GSR, a reconfigurable on-chip *LC* resonant clock distribution solution, was shown. This generalized series resonance (GSR) technique can achieve 50% driver power savings compared to non-resonant drivers, while reducing the skew by 50% (below 10ps) to make it easier to achieve timing closure. Taking processor designs as a benchmark 25% power and 25% area can be assumed to be consumed by CDN for an NR design. A 25% reduction in clock power can result in more than 6% savings in the overall power. At the worst case there can be 5% increase in die costs which can be compensated by yield gain from timing margins. Decrease in hot spots can increase the reliability and more than 5% increase in the life time of the ICs.

Thus, recycling energy in this fashion reduces the hotspot occurrences that were discussed in the motivation section 1.1. All these can lead to much lower cooling costs for workstation and server farms increasing their reliability and leading to more sustainable IT infrastructure.

The key performance index energy-delay product, which is usually lowered with ‘More of Moore’ technology scaling, is shown to be improved through a ‘More than Moore’ solution using inductors.

The power reduction solutions presented in this thesis do entail an enhancement in the design flow and development of CAD software for automatic inductor synthesis. These are one-time costs that are far less than the typical development costs of current DSM SoCs and processors.

11.3 Future Work

Using the equations derived, further work is now possible to automatically synthesize GSR and PSR solutions with power and timing optimization. Further work is now possible to develop automatic place and route (APR) solutions to synthesize series resonance solutions, thus allowing their main stream deployment. Various GSR configurations can be fabricated on test chip to verify the theoretical predictions. Once these unit cells are characterized and incorporated into the standard cell library data base, main stream applications can be addressed.

Future work will address optimal layout implementation of GSR with multiple inductors and distributed parasitics for power and delay optimizations in asymmetric trees. An actual clock tree from low power processor like ARM can be taken and converted into a resonant based driver and distribution scheme. Various resonance schemes can be applied at multiple levels of the clock distribution hierarchy. Data paths can be converted to dynamic logic scheme to save overall power. Most of the inductors in data path can be shared between various lines.

Statistical Static timing analysis can be better applied to PSR and a far better PPA optimization can be obtained with improved yields. The use of inductors also

opens the possibility of using injection locking techniques to improve the jitter in clocks [48].

This work further advances the cause of using energy saving resonance in future SoCs and processors by providing new topologies and a comprehensive trade-off analysis for the first time.

12 REFERENCES

[1] P. Restle, D. Shan, D. Hogenmiller, Y. Kim, A. Drake, J. Hibbeler, T. Bucelot, G. Still, K. Jenkins and J. Friedrich2, "Wide-Frequency-Range Resonant Clock with On-the-Fly Mode Changing for the POWER8™ Microprocessor," in *IEEE International Solid-State Circuits Conference*, 2014, pp. 100-101.

[2] T. N. Theis and P. M. Solomon, "In quest of the Next Switch: Prospects for greatly reduced power dissipation in a successor to the silicon field-effect transistor," *Proc. IEEE*, vol. 98, no. 12, pp. 2005–2014, Dec. 2010.

[3] X.Hu and M. Guthaus, "Distributed LC Resonant Clock Grid Syntheis," *IEEE Transactions on Circuits And Systems—I: Regular Papers*, Vol. 59, No. 11, pp. 2749-2760, November 2012.

[4] L. Chang, D. Frank, R. K. Montoye, S. J. Koester, B. L. Ji, P. W.Coteus, R. H. Dennard, and W. Haensch, "Practical strategies for power-efficient computing technologies," *Proc. IEEE*, vol. 98, no. 2, pp. 215–236, Feb. 2010.

[5] Shien-Yang Wu, C.Y. Lin, S.H. Yang, J.J. Liaw and J.Y. Cheng, "Advancing foundry technology with scaling and innovations," in *Proc, International Symposium on VLSI-TSA*, 2014, pp. 1-3

[6] H. A. Moore, M. Dietrich, A. Herkersdorf, F. Miller, T. Wild, K. Hahn, A. Grunewald, R. Bruck, S. Krohnert and J. Reisinger, "System integration — The bridge between More than Moore and More," in *Proc. DATE*, 2014, pp. 1-9.

[7] R. K. Jana, G. L. Snider and D. Jena, "Energy-Efficient Clocking Based on Resonant Switching for Low-Power Computation," *IEEE Transactions On Circuits And Systems—I: Regular Papers*, Vol. 61, No. 5, pp. 1400-1408, May 2014.

[8] A. Zolfaghari, A. Chan, and B. Razavi, "Stacked inductors and transformers in CMOS technology," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 620–628, Apr. 2001.

[9] K. Suhwan, C. Ziesler, M. Papaefthymiou, "Charge-recovery computing on silicon," *IEEE Transactions on Computers*, vol. 54 , issue 6, pp. 651- 659, June 2005.

[10] Yibin Ye and K. Roy "Energy recovery circuits using reversible and partially reversible logic*," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Volume: 43, Issue: 9, pp. 769-778, 1996.

[11] V. S. Sathe, V. Arekapudi, C. Ouyang, M. Papaefthymiou, A. Ishii and S. Naffziger, "Resonant-Clock Design for a Power-Efficient, High-Volume x86-64 Microprocessor. *IEEE J. Solid-State Circuits*, vol. 48, no.1, pp.140–149, Jan. 2013.

[12] S. C. Chan, K. L. Shepard, and P. J. Restle, "Uniform-phase, uniform amplitude, resonant-load global clock distributions," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 102–109, Jan. 2005.

[13] J. Rosenfeld and E. Friedman, "Design methodology for global resonant H-tree clock distribution networks," *IEEE Transactions on Very Large Scale Integration (VLSI) systems*, vol. 15, no. 2, pp. 135–148, February 2007.

[14] S. C. Chan, P. J. Restle, T. J. Bucelot, J.S. Liberty, S. Weitzel, J. M. Keaty, B. Flachs, R. Volant, P. Kapusta, and J. S. Zimmerman, "A Resonant Global Clock Distribution for the Cell Broadband Engine Processor," *IEEE Journal Of Solid-State Circuits*, Vol. 44, No. 1, pp. 64-72, January 2009.

[15] T. Lee, "passive RLC networks," in *The design of CMOS RF integrated circuits*, New York: Springer.

[16] Yuhui Chen, F. Lee, L. Amoroso and H. Wu, "A resonant MOSFET gate driver with efficient energy recovery," *IEEE Transactions On Power Electronics*, Vol. 19, No. 2, pp. 470- 477, March 2004.

[17] N. Kurd, M. Chowdhury, E. Burton, T.P. Thomas, C. Mozak, B. Boswell, M. Lal, A. Deval, J. Douglas, M. Elassal, A. Nalamalpu, T.M. Wilson, M. Merten, S. Chennupaty, W. Gomes, R. Kumar, "Haswell: A family of IA 22nm processors," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2014, pp. 112-113.

[18] M. Guthaus, G. W. Silke and R. Reis, "Revisiting Automated Physical Synthesis of High-Performance Clock Networks," *ACM Transactions on Design Automation of Electronic Systems,"* Vol. 18, No. 2, Article 31, pp.31:1-31:2, March 2013

[19] A. Ishii, J. Kao, V. Sathe, and M. Papaefthymiou, "A resonant-clock 200MHz ARM926EJ-S™ microcontroller," in *Proc. IEEE Eur. Solid-State Circuits Conf.* , Sep. 2009, pp. 356–359.

[20] Alan J. Drake, Kevin J. Nowka, Tuyet Y. Nguyen, Jeffrey L. Burns, and Richard B. Brown, "Resonant Clocking Using Distributed Parasitic Capacitance," *IEEE Journal Of Solid-State Circuits*, Vol. 39, No. 9, pp. 1520-1528, Sept 2004.

[21] H Mahmoodi, V. Tirumalashetty, M. Cooke, and K. Roy, "Ultra Low-Power Clocking Scheme Using Energy Recovery and Clock Gating," *IEEE Transactions On Very Large Scale Integration Systems*, Vol. 17, No. 1, pp. 33-44, January 2009.

[22] V. Sathe, "Hybrid resonant-clocked digital design," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Univ. of Michigan, Ann Arbor, MI, May 2007.

[23] I. Bezzam and S. Krishnan, "A Pulsed Resonance Clocking for Energy Recovery," in *Proc. IEEE International Symposium on Circuits and Systems*, Melbourne, 2014, pp. 2760 - 2763.

[24] H. Fuketa, M. Nomura, M. Takamiya and T. Sakurai, "Intermittent Resonant Clocking Enabling Power Reduction at Any Clock Frequency for Near/Sub-Threshold Logic Circuits" *IEEE J. Solid-State Circuits*, vol. 49, no. 2, pp. 536– 544, Feb. 2014.

[25] K. Ikeuchi, K. Sakaida, K. Ishida, T. Sakurai and M. Takamiya, "Switched Resonant Clocking (SRC) scheme enabling dynamic frequency scaling and low-speed test," *Custom Integrated Circuits Conference*, 2009, pp. 33- 36.

[26] I. Bezzam, C. Mathiazhagan, S. Krishnan and T. Raja, "Low Power Low Voltage Wide Frequency Resonant Clock and Data Circuits for SoC Power Reductions," *IEEE Latin American Symposium on Circuits and Systems*, Peru, February 2013.

[27] J. M. Rabaey, A. Chandarakasan and B. Nokolic, *Digital Integrated Circuits: A Design Perspective*, 2nd Ed. New Jersey: Prentice Hall, pp. 349-361, 2003.

[28] J. Rabaey, "Optimizing Power @ Design Time- Circuit Level Techniques" in *Low Power Design Essentials*, 1st Ed. New York: Springer, 2009, pp. 86-88.

[29] G. Wilke, R. Fonseca, C. Mezzomo, and R. Reis, "A novel scheme to reduce short-circuit power in mesh-based clock architectures," in *Proc. SBCCI*, 2008, pp. 117–122.

[30] C. Yoo, "A CMOS buffer without short-circuit power consumption," *IEEE Trans. Circuits System II, Analog Digit. Signal Process*, vol. 4, no. 9, pp. 935–937, Sep. 2000

[31] S. E. Esmaeili, A. J. Al-Kahlili, and G. E. R. Cowan, "Low-Swing Differential Conditional Capturing Flip-Flop for LC Resonant Clock Distribution Networks," *IEEE Transactions On VLSI Systems*, Vol. 20, No. 8, pp.1547-1551, August 2012.

[32] C. N. Sze, P. Restle, G.-J. Nam, and C. J. Alpert, "Clocking and the ISPD'09 clock synthesis contest," in Proc. ISPD, 2009, pp. 149–150

[33] S. Esmaeili, A. Al-Khalili, and G. Cowan, "Dual-edge triggered sense amplifier flip-flop for resonant clock distribution," *IET Computers & Digital Techniques*, vol. 4, no. 6, pp. 499 - 514, 2010.

[34] S. E. Esmaeili, A. J. Al-Kahlili, and G. E. R. Cowan, "Estimating Required Driver Strength in the Resonant Clock Generator," *IEEE Transactions On VLSI Systems*, Vol. 20, No. 8, pp.927-930, August 2012.

[35] C. Yue and S. Wong, "On-chip spiral inductors with patterned ground shields for Si-based RF ICs," *IEEE J. Solid-State Circuits*, vol. 33, no.5, pp. 743–752, May 1998.

[36] Arizona State Univerisity, Predictive technology models (PTM): http://ptm.asu.edu/

[37] W. Zhao, Y. Cao, "New generation of Predictive Technology Model for sub-45nm early design exploration," *IEEE Transactions on Electron Devices*, vol. 53, no. 11, pp. 2816-2823, November 2006.

[38] B. Razavi, Chapter 2 in *Design of Analog CMOS Integrated Circuits*, 1st ed. New York, NY, USA: McGraw-Hill Higher Education, 2000.

[39] W. Daly, *Digital Systems Engineering*, 2nd Ed. New Jersey: Prentice Hall, 2003, pp. 349-361.

[40] Domenico Campolo, Metin Sitti and Ronald S. Fearing, "Efficient Charge Recovery Method for Driving Piezoelectric Actuators with Quasi-Square Waves," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 50, no. 1, pp.1-9, January 2003.

[41] I. Bezzam and S. Krishnan, "Minimizing Power and Skew in VLSI-SoC Clocking With Pulsed Resonance Driven De-skewing Latches," in *IEEE 27th International Conference on VLSI Design*, 2014, pp. 157-161.

[42] C. Kim and S. Kang (2002). A Low-Swing Clock Double-Edge Triggered Flip-Flop. *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 5, May 2002, pp. 648-652.

[43] J. Tschanz, S. Narendra, Z. Chen, S. Borkar, and M. Sachdev (2001). Comparative Delay and Energy of Single Edge-Triggered & Dual Edge-Triggered Pulsed Flip-Flops for High-Performance Microprocessors. *Proceedings of 2001 ISLPED*, pp. 147-152, August 6-7, 2001, USA.

[44] Terence M. Potter and James Blomgren (2006). Null value propagation for FAST14 logic. *US patent* No. 7,053,664, May 2006.

[45] I. Bezzam, S. Krishnan and C. Mathiazhagan (2012). Low power SoCs with Resonant Dynamic Logic using Inductors for Energy Recovery. VLSI-SoC.

[46] Chirayu S. Amin, Noel Menezes, Kip Killpacks, Florentin Dartus, Umakanta Choudhug, Nagib Hakims, Yehea I. Ismail "Statistical Static Timing Analysis: How simple can we get?" DAC2005, June l3-17,2005, Anaheim, California, USA.

[47] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba (2011). *Advanced Configuration and Power Interface (ACPI) is an open industry specification 5.0*: http://www.acpi.info

[48] Zheng Xu and Kenneth L. Shepard "Design and Analysis of Actively-Deskewed Resonant Clock Networks," *IEEE Journal Of Solid-State Circuits*, Vol. 44, No. 2, February 2009 pp. 558-568.

Nomenclature

C	Capacitor
CDN	Clock Distribution Network
C_L	Load Capacitor
CMOS	Complementary Metal Oxide Semiconductor
C_{OUT}	Output Capacitor
CPR	Continous Parallel Resonance
D	Data input of a flip-flop
DC	Direct Current
DCR	DC resistance of inductor
DDR	Double Data Rate
DET	Dual Edge Triggering
DVFS	Dynamic Voltage Frequency Scaling
E_C	Energy stored on capacitor C per cycle
EMI	Electro-Magnetic Interference
ESR	Electrical Series Resistance of Capacitor
E_{VDD}	Energy drawn from V_{DD} supply per cycle
f_{CLK}	Clock Frequency
f_R	Frequency of damped oscillations
f_{RES}	ideal Frequency of Resonance
GSR	Generalized Series Resonance
IC	Integrated Circuit
i_L	Inductor Current
INV	Standard medium Inverter driving 1pF load

IR	Intermittent Resonance
L	Inductor
LC	Inductor (L) Capacitor (C) series/parallel combination
LCB	Local Clock Buffers
MEMS	Micro-Electro-Mechanical Systems
MS	Master Slave
NEMS	Nano-Eleectro-Mechanical Systems
NMOS	N-type Metal Oxide Semiconductor
NR	No Resonance
P_{avg}	Average Power per cycle
P_{CPR}	CPR Power
P_{GSR}	GSR Power
PLS_CLK	Clock Pulse Stream
PMOS	P-type Metal Oxide Semiconductor
P_{NR}	Non Resonant Power
PPA	Power, Performance and Area
P_{PSR}	PSR Power
PSR	Pulsed Series Resonance
Q (italicized)	Quality factor
Q	Output of flip-flop
Q_C	Component Quality factor of Capacitor *C*
Q_L	Component Quality factor of Inductor *L*
R_d	pull-Down switch Resistance
RF	Radio Frequency
R_p	Inductor parallel Resistance equivalent to DCR

R_r	Resonance on-off switch Resistance
R_u	pull-Up switch Resistance
R_w	Interconnect Wire Resistance
SCB	Sector Clock Buffers
SoC	System on Chip
T_{CLK}	Clock Period
T_{PW}	Pulse Width Time
TSPC	True Single Phase Clocking
V_C	Capacitor Voltage
V_{DD}	Power Supply voltage connected to Drain of PMOS
V_{in}	Input Voltage
V_{LB}	Inductor Bias Voltage
V_{OH}	logic Output High Voltage
V_{OL}	logic Output Low Voltage
V_{OUT}	Output Voltage
μ	micro meter units
τ	time constant

Appendix A: *MATLAB for solving ODE and Deriving Expressions*

A- 1 Power in CPR

Integrating V^2/R averaged over period T

```
>> syms Vdd t Tr Fr x
>> syms R Q Fr Fc
```

At resonance

```
>> y=int((.5*Vdd+.5*Vdd*sin(2*pi*t/Tr))^2,0,Tr)*pi*2*Q*Fr*C/(Tr)
```

y = (3*pi*C*Fr*Q*Vdd^2)/4

➔ **Same as hand derivation**

At non resonance Fc = x. Fr

```
y=int((.5*Vdd+.5*Vdd*sin(2*pi*t/Tr))^2,0,Tr/x)*pi*2*Q*x*Fc*C/(Tr/x)
 y =(C*Fc*Q*x*(12*pi*Vdd^2 + 8*Vdd^2*x - 8*Vdd^2*x*cos((2*pi)/x) -
Vdd^2*x*sin((4*pi)/x)))/16

=C*Fc*Q*Vdd^2 - C*Fc*Q*Vdd^2*cos(pi/x)^2 + (3*pi*C*Fc*Q*Vdd^2)/(4*x) -
(C*Fc*Q*Vdd^2*cos(pi/x)^3*sin(pi/x))/2 + (C*Fc*Q*Vdd^2*cos(pi/x)*sin(pi/x))/4
```

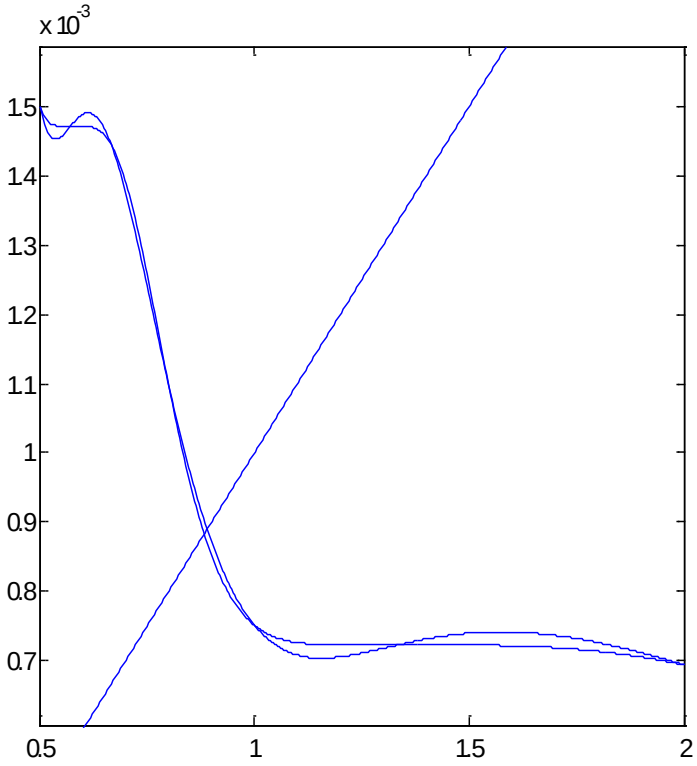
>>> Example

```
Fc =   1.0000e+09
Q =    3.1400
Vdd =     1
>> eval(z)
 ans =((3*pi)/10 + x/5 - (x*cos((2*pi)/x))/5 - (x*sin((4*pi)/x))/40)/(1256*x)

 >> expand(z)
 ans = (cos(pi/x)*sin(pi/x))/12560 - cos(pi/x)^2/3140 + (3*pi)/(12560*x) -
(cos(pi/x)^3*sin(pi/x))/6280 + 1/3140
zz=(3*pi)/(12560*x)  + 1/3140 - cos(pi/x)^2/3140
```

>>> Comparing Results from SPICE

```
>> hold off
>> ezplot(zz, [0.5,2])
>> hold on
>> ezplot(n, [0.5,2])
>> ezplot(z, [0.5,2])
```

Approximation of closed form Power vs. Frequency close to sims



A- 2 PSR Evaluating expressions for V_{OL} & V_{OH}

PSR V_{OL} derivation
vv(t) = vv(t)=.5*Vdd+.5*Vdd*exp(-t*pi/(Tr*Q))*cos(2*pi*t/Tr)
>> eval(vv(Tr/2))

ans = Vdd/2 - (Vdd*exp(-pi/(2*Q)))/2

- ➔ **Same as hand derivation**
- ➔ **PSR V_{OH} derivation**
- ➔ >> eval(vv(Tr))
- ➔
- ➔ **ans = Vdd/2 + (Vdd*exp(-pi/Q))/2**

- ➔ **Same as hand derivation**

A- 3 Ordinary Differential Equations (ODE) Solving PSR

First order example with initial conditions

```
>> syms u(t)
> Du=diff(u);
>> dsolve(diff(u,2)==u,u(0)==1,Du(0)==0) l
ans =exp(-t)/2 + exp(t)/2
```

Solving PSR Differential Equation

```
>> syms C L R Vdd V
>> V=dsolve(diff(u,2)==-diff(u)/C*R -u/(L*C),u(0)==0,Du(0)==Vdd/2*R*C )
simplify(V)
 ans =-(C^2*L*R*Vdd*exp(-(t*((L^2*R^2 - 4*C*L)^(1/2) + L*R))/(2*C*L)) -
C^2*L*R*Vdd*exp((t*((L^2*R^2 - 4*C*L)^(1/2) - L*R))/(2*C*L)))/(2*(L^2*R^2 -
4*C*L)^(1/2))

>> I=dsolve(diff(u,2)==-diff(u)/tLR -u/(tLC*tLC),u(0)==1)
 I =C4*exp(-(t*(tLC + ((tLC - 2*tLR)*(tLC + 2*tLR))^(1/2)))/(2*tLC*tLR)) - exp(-
(t*(tLC - ((tLC - 2*tLR)*(tLC + 2*tLR))^(1/2)))/(2*tLC*tLR))*(C4 - 1)
>> VL(t)=int(I)
(tLC*exp(- t/(2*tLR) - (t*(tLC^2 - 4*tLR^2)^(1/2))/(2*tLC*tLR))*(2*C4*tLR^2 -
C4*tLC^2 - 2*tLR^2*exp((t*(tLC^2 - 4*tLR^2)^(1/2))/(tLC*tLR)) +
2*C4*tLR^2*exp((t*(t

>> solve(VL(0)/CL-Vdd/2,C4)
 ans =(4*tLC*tLR^2 - CL*Vdd*tLR*(tLC^2 - 4*tLR^2)^(1/2) +
CL*Vdd*tLC*tLR)/(8*tLC*tLR^2 - 2*tLC^3 + 2*tLC^2*(tLC^2 - 4*tLR^2)^(1/2))

>> Vo(t)=int(0.5*Vdd*((W0/Wd)^2)*exp(-a*t)*sin(Wd*t))
Vo(t) =-(Vdd*W0^2*exp(-a*t)*(Wd*cos(Wd*t) + a*sin(Wd*t)))/(2*Wd^2*(Wd^2 +
a^2))
```

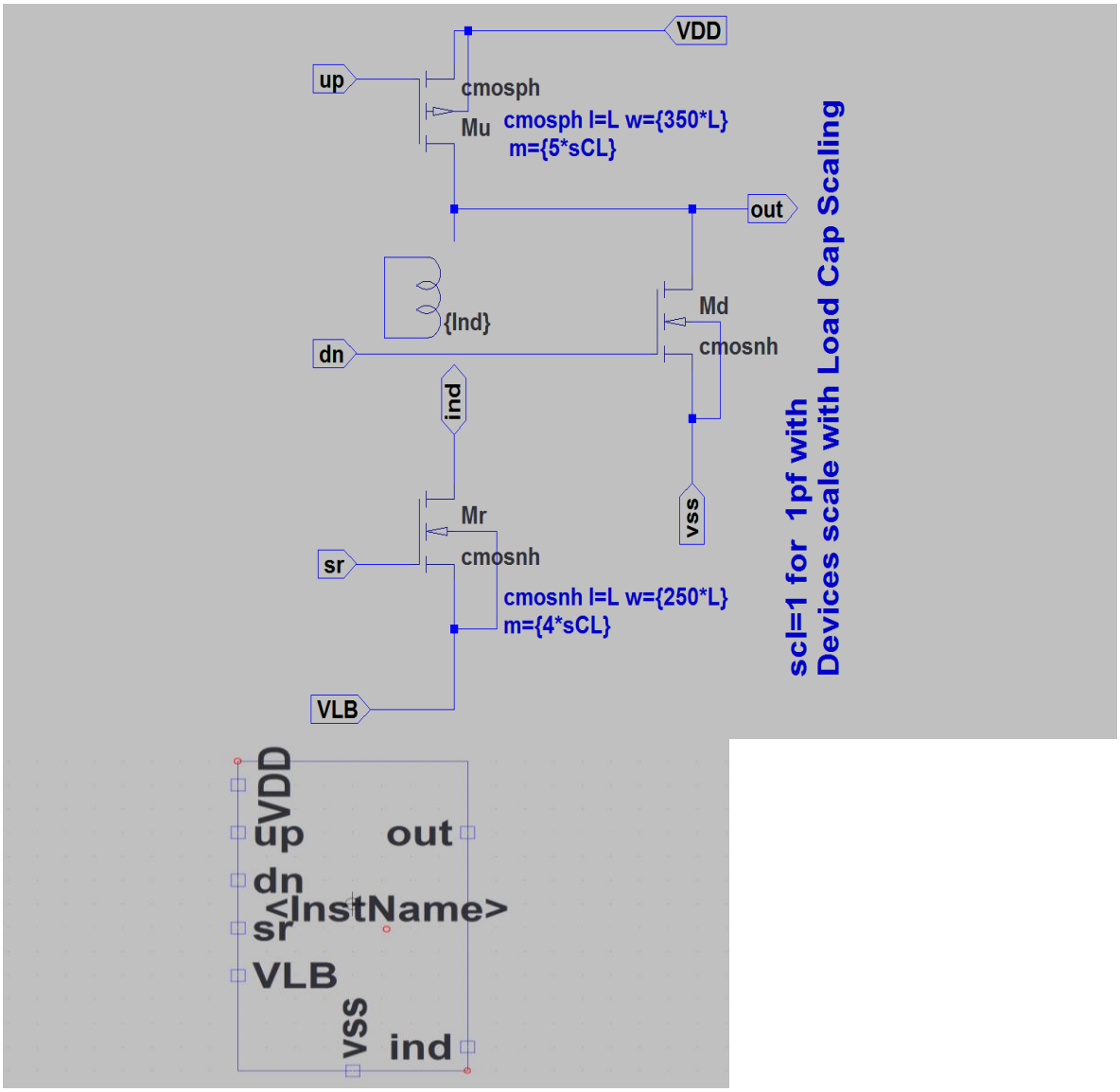
Delay Calculations

```
>> V(t)=0.5*Vdd+0.5*(Vdd*exp(-R*t/(2*L))*cos(t/sqrt(L*C)))

V(t) = Vdd/2 + (Vdd*exp(-(R*t)/(2*L))*cos(t/(C*L)^(1/2)))/2

>> solve(V(t)==Vdd*0.5,t)
```

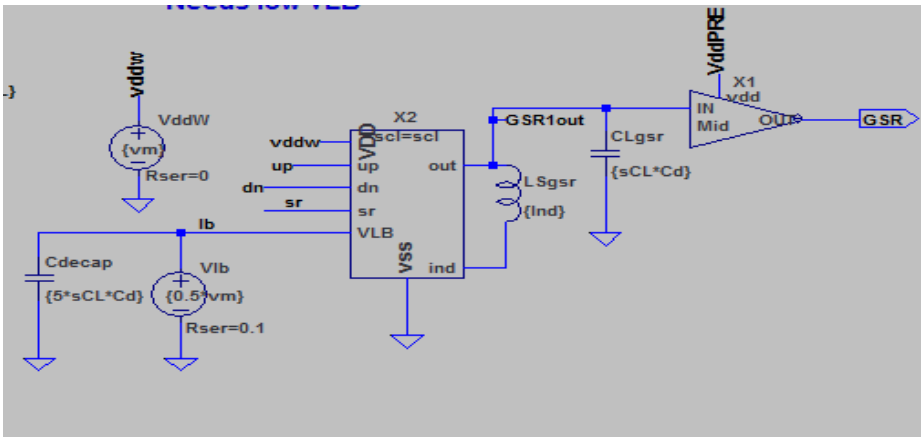
ans = (pi*(C*L)^(1/2))/2

➔ **Same as hand derivation= Tres/4**

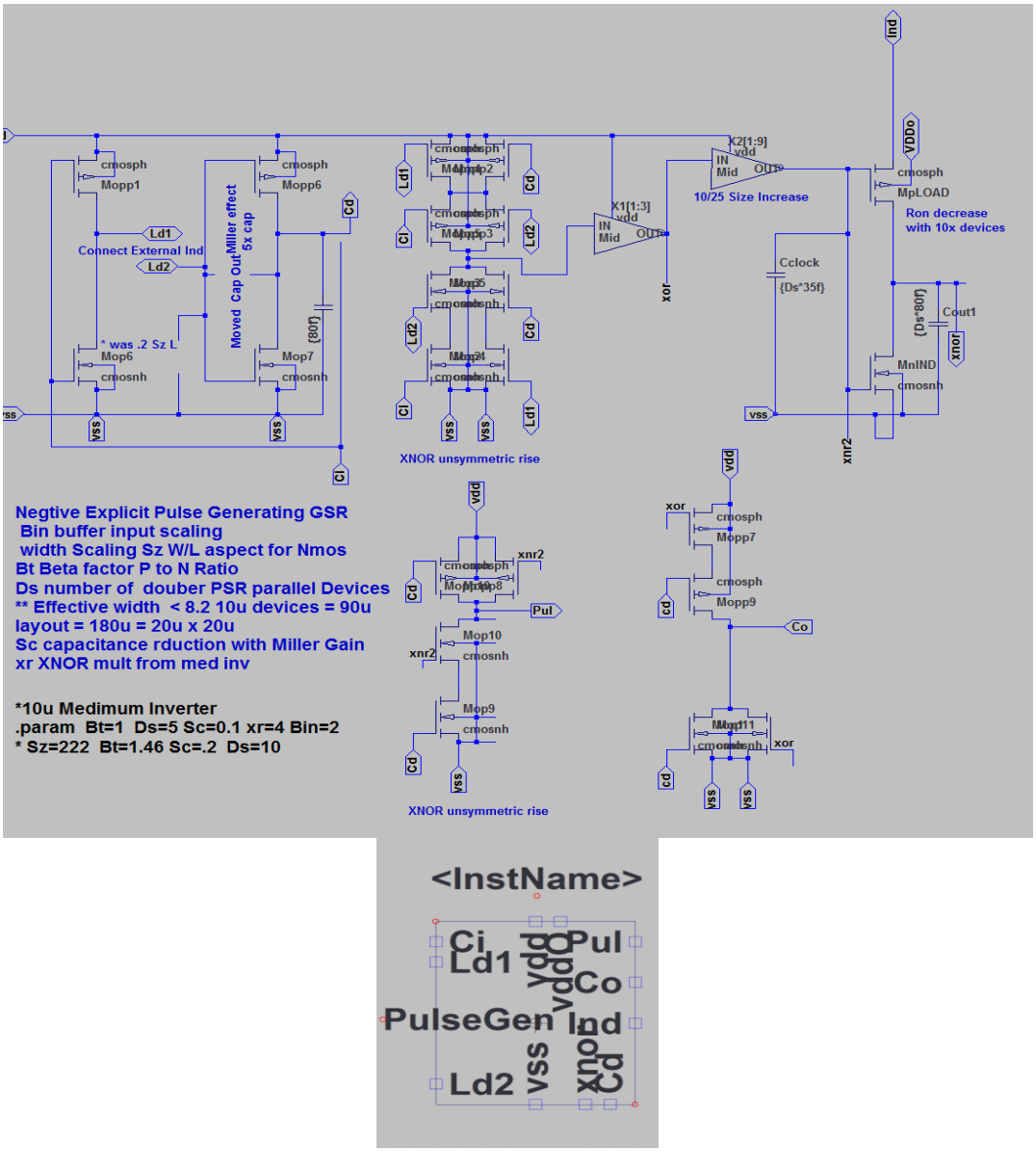
Appendix B: *LTSPICE Schematic Diagrams*



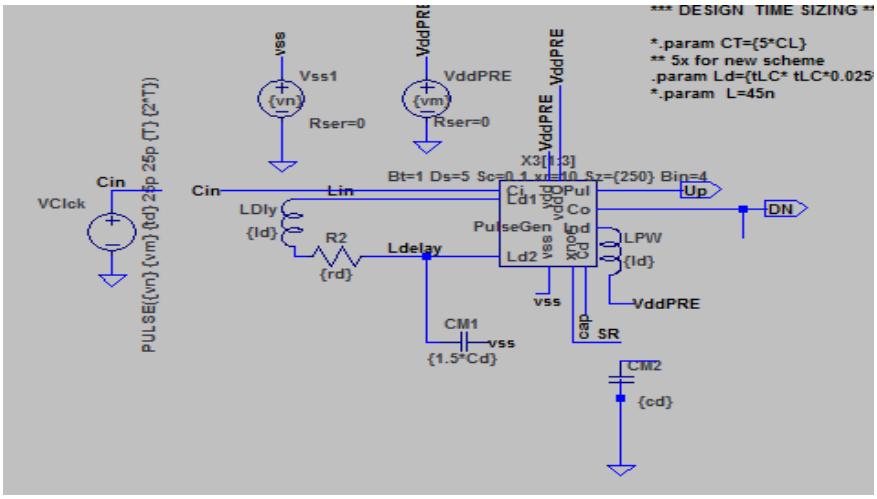
B 1 GSR Scalable Reconfigurable Driver Schematic and Macro Cell Symbol



B 2 Typical Configuration of Driver for GSR rail to rail operation

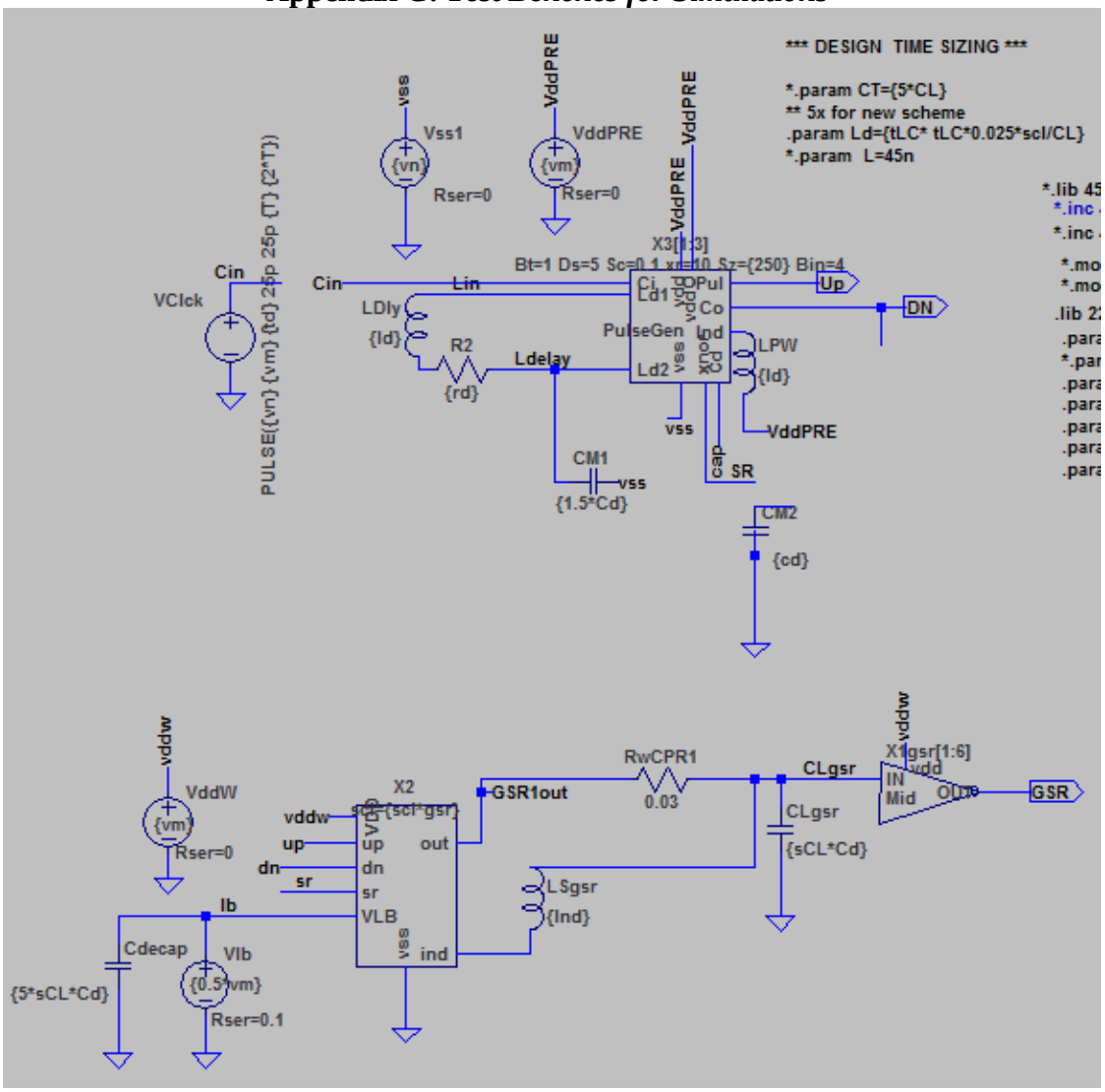


B 3 GSR Scalable Predriver and Symbol

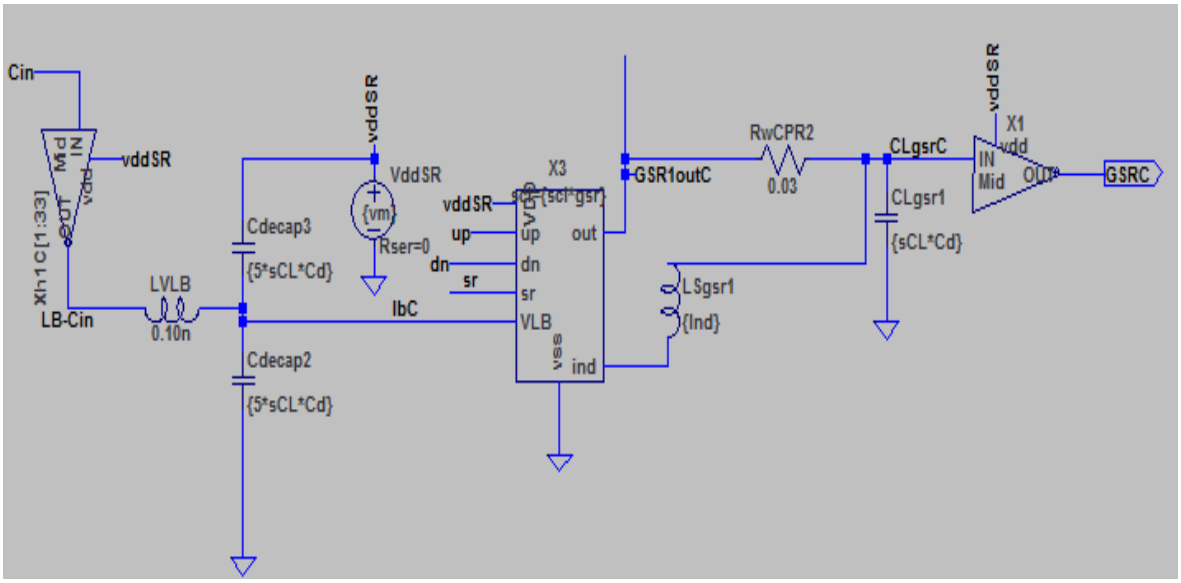


B 4 Typical Configuration of Predriver

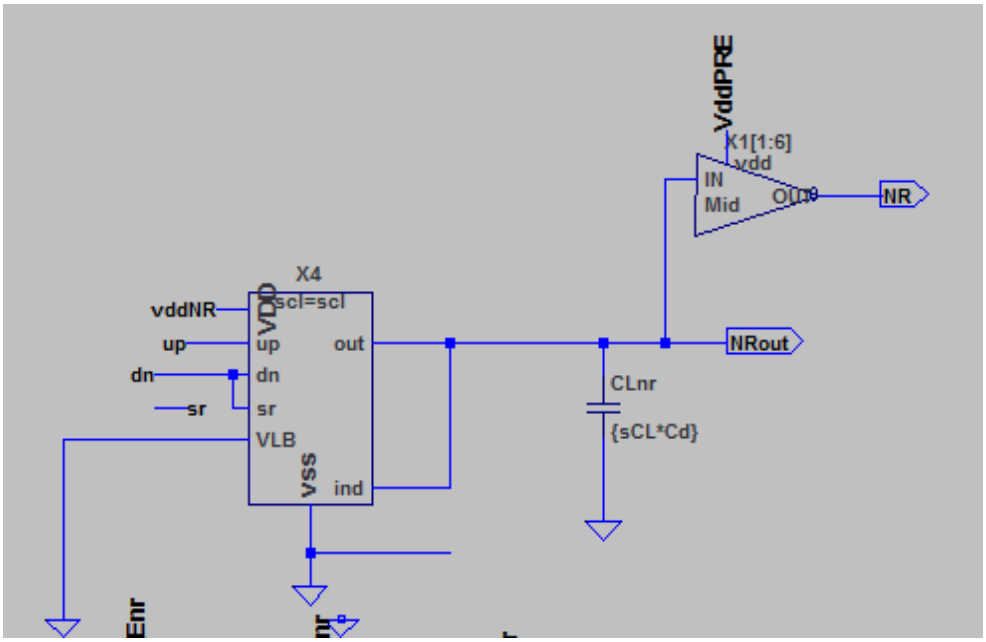
Appendix C: *Test Benches for Simulations*



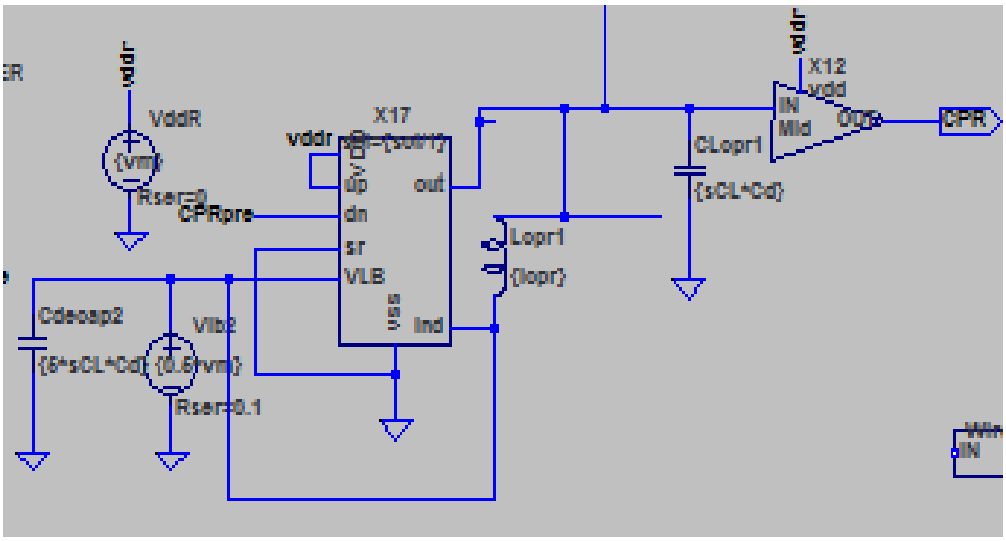
C- 1 Test bench of GSR configuration with Predriver and bias voltage for inductor



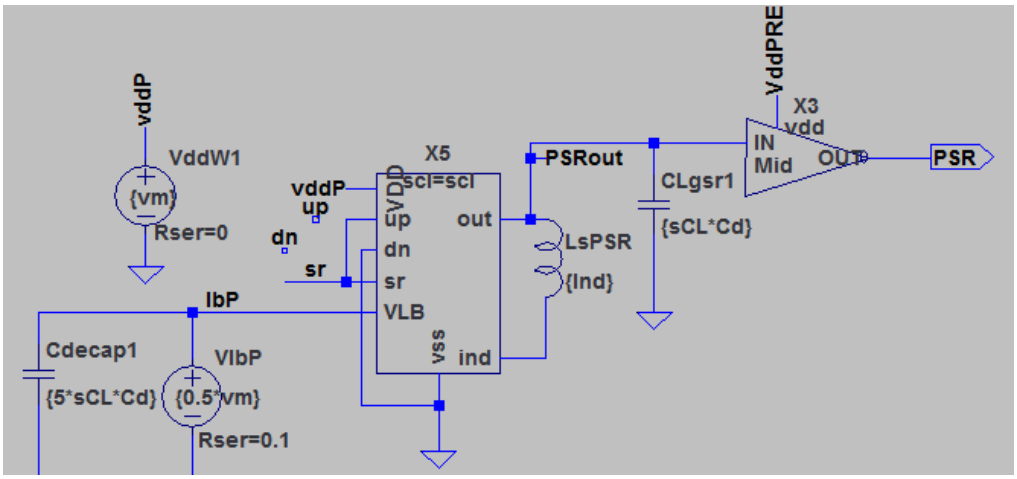
C- 2 Test bench with GSR using external capacitor to generate inductor bias (GSR-C)



C- 3 NR configuration with GSR macro cell



C- 4 CPR Configuration with GSR macro cell



C- 5 PSR configuration with GSR macro

Appendix C: *Spread Sheet for Design Calculations*

Available as Design Aids at: tinyurl.com/Bezzam

Calculate	Qc	Cp pF	ESR Ω	QL	Ls nH	rS Ω	Fres GHz	Qtnk	Fr GHz	Tr nS
ESR,rS, Qt, Fr	30	1	4.77	3.14	0.5	1.0	1	3.872	**0.992**	
Fres, ESR, rS, Qt, Fr	30	1	0.95	3.14	1	10.1	5.0	2.870	**4.956**	**0.2**
Ls,rS, Qt, Fr ESR,	30	20	0.24	3.14	1.27	**2.5**	1	2.870	**0.985**	
Fres, ESR, rS, Qt, Fr	30	20	0.24	3.14	1.25	2.5	1.0	2.870	**0.991**	
Ls,rS, Qt, Fr ESR,	30	20	0.05	3.14	0.05	**0.5**	5	2.870	**4.924**	**0.2**
Ls,rS, Qt, Fr ESR,	30	160	0.03	3.14	0.16	**0.3**	1	2.870	**0.985**	
Ls,rS, Qt, Fr ESR,	30	160	0.01	3.14	0.006	**0.1**	5	2.870	**4.924**	**0.2**

Highlighted items show derived values.

Red items are for calculated critical parameters

Appendix D: ***Design Synthesis Algorithms***

Algorithm C1: Resonant Grid Generation
Input: Near-zero skew routed tree without buffers
Output: Routed tree with resonant local sectors
1: Insert min-size Local Clock Buffer (LCB) at root
2: Place LC Tank at output of LCB
3: Size LC tank ***(C2)***
4: Adjust LC Tank Placement ***(C3)***
5: while Voltage swing at sinks < 90% do
6: Increase Buffer size
7: Size LC tank
8: Run Spice sims to verify swing
9: end while

Algorithm C2: inductor placement and sizing algorithm
Input: Near-zero skew routed tree with LCB at root & Grid nodes from A-I, f_{CLK} ; Lmin-max, MA-max (inductor metal area); Skew t_{skw} constraint
Output: Inductor sizes and locations
Output: Correctly sized LC tank for resonance at the desired frequency
1: Ltank = 1/ω²C
2: Run Spice
3: while | fdesired − fmin| > 10MHz do
4: L = L− |fdesired−fmin|/ fmin
5: Run Spice

Algorithm C3: inductor placement and sizing algorithm
Input: Properly sized tank, topologically sorted list of nodes in tree
Output: LC Tank placed at a node that provides good voltage swing
1: maxSwingNode = Null
2: minSwing = 0
3: Remove tank from LCB output.
4: for n 2 First 10% of nodes in tree do
5: Place tank at n.
6: Run Spice
7: if min V(sinks) > V(minSwing) then
8: maxSwingNode = n
9: minSwing =min V(sinks)
10: end if
11: Remove tank from n.
12: end for
13: Place tank at maxSwingNode