



# Design Automation of Series Resonance Clocking in 14-nm FinFETs

Dhandeep Challagundla<sup>1</sup> · Ignatius Bezzam<sup>2</sup> · Riadul Islam<sup>1</sup>

Received: 16 February 2023 / Revised: 13 July 2023 / Accepted: 14 July 2023 /

Published online: 1 August 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Power-performance constraints have been the key driving force that motivated the microprocessor industry to bring unique design techniques in the past two decades. The rising demand for high-performance microprocessors increases the circuit complexity and data transfer rate, resulting in higher power consumption. This work proposes a set of energy recycling resonant pulsed flip-flops to reuse some of the dissipated energy using series inductor–capacitor (LC) resonance. Moreover, this work also presents wideband clocking architectures that use series LC resonance and an inductor tuning technique. By employing pulsed resonance, the switching power dissipated is recycled back. The inductor tuning technique aids in reducing the skew, increasing the robustness of the clock networks. This new resonant clocking architecture saves over 43% power and 90% reduced skew in clock tree networks and saves 44% power and 90% reduced skew in clock mesh networks, clocking a range of 1–5 GHz frequency, compared to conventional primary–secondary flip-flop-based clock networks. Implementation of resonant clock architectures on standard clock network benchmarks depicts 66% power savings and 6.5× reduced skew while using the proposed pulsed resonant flip-flop and saves 64% power and 12.7× reduced skew while using the proposed resonant true single-phase clock (TSPC) flip-flop.

**Keywords** Clock skew · LC resonance · Clock tree architecture · Pulsed flip-flops · Power consumption

---

✉ Riadul Islam  
riadulu@umbc.edu

Dhandeep Challagundla  
vd58139@umbc.edu

Ignatius Bezzam  
i@rezonent.us

<sup>1</sup> Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD 21250, USA

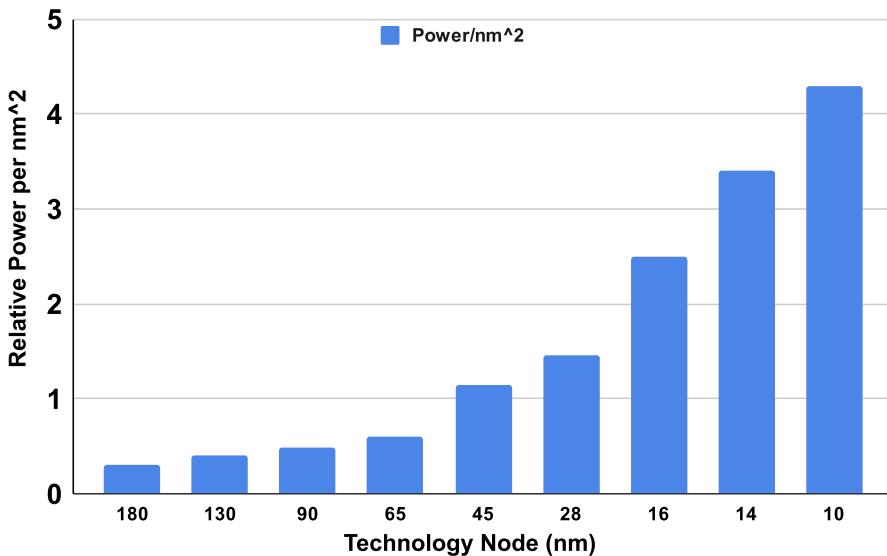
<sup>2</sup> Rezonent Inc., 1525 McCarthy Blvd, Milpitas, CA 95035, USA

## 1 Introduction

Power consumption is one of the major problems faced in the high-performance microprocessor industry [14, 35, 50]. The need for an increase in performance has steered the operating frequencies higher, resulting in an increased complexity among the microprocessor designs [29, 30, 50]. Figure 1 shows the power consumption per  $\text{nm}^2$  trend compared with technology scaling over the past two decades. The relative power consumed per  $\text{nm}^2$  increases exponentially as the technology scales down. This higher power led designers to constantly come up with innovative techniques to reduce the power while trying to meet all the design constraints that impact the performance [8, 20, 23, 25, 28, 31, 41, 48, 52].

Multiple regions of high-performance microprocessors, such as memory, logic cells, and clock network, are sources of high power consumption. A significant portion of dynamic power consumed in a high-frequency design is due to the switching activity in the clock network [37]. Researchers have developed several low-power techniques to reduce active power consumption in clock networks. Among them, inductor-based LC resonant clocking techniques have great potential to save switching power due to their constant phase and magnitude. However, most industry-standard electronic design automation (EDA) tools do not explicitly support integrating LC resonance in the clock architectures. Additionally, designing a resonant clock architecture requires the designer to have multiple domain expertise due to the nonlinear behavior of inductors.

To enable a resonant clock architecture, we need resonant flip-flops (FFs) for synchronous circuits. However, they occupy a substantial chip area consuming high power. Researchers proposed many low-power flip-flops, however, not suitable for resonant



**Fig. 1** As the technology scales down, the relative power consumed per  $\text{nm}^2$  increases exponentially as the density of the transistors goes high [16]

operations [6, 11, 32, 44, 45, 49]. In this research, we propose several conventional register-based pulsed FFs suitable for series resonance and reduce the overall power consumption in the clock network.

Besides power, skew plays a critical role in enabling high-frequency operation. To reduce the skew generated by clock networks, we introduce an inductor tuning technique to match the series resonance inductor with the load capacitance of the clock network, which produces pulses with equal resonant frequencies. Although the resonant frequency depends on the series inductor and the load capacitance, we generate a constant pulse width for a wideband (WB) of input clock frequencies. Therefore, calibrating the inductors once for a clock network architecture enables WB frequency operation.

## 1.1 Main Contributions

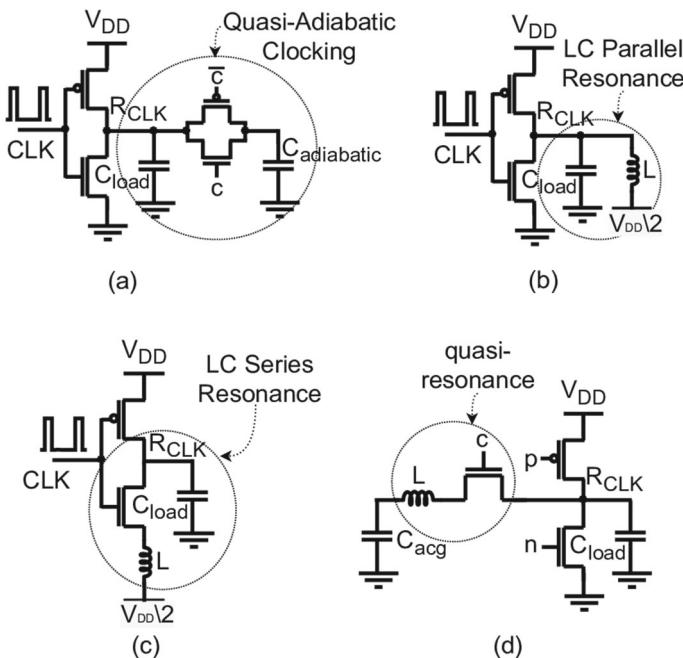
This work proposes a clocking architecture to recycle the power dissipated by the synchronous elements using series resonance technology and balance clock skew by adapting the proposed inductor tuning technique. In particular, the main contributions of this work are:

- Clocking architectures to recycle the power dissipated by synchronous elements using series resonance.
- A novel matching pulse generator with dual-rail booster using series resonance.
- A set of pulsed register-based FFs exploits the behavior of pulsed series resonance.
- Demonstration of inductor tuning technique algorithms to compensate for the skew in the clock tree architectures on industrial testbenches.

The rest of the document is organized as follows. Chapter 2 discusses several energy recycling clocking techniques and low-power flip-flop designs. The proposed low-power resonant flip-flops are described in Chapter 3. Chapter 4 discusses the proposed wideband resonant clocking architectures that utilize a boosted-amplitude pulse signal generated from the system clock signal. The skew reduction methodology, described in Chapter 5, explains the inductor tuning technique that results in reduced skew, followed by algorithms to automate the proposed inductor tuning technique. The implementation results of the proposed wideband resonant clocking architectures, shown in Chapter 6, support our claim of power and skew reduction. The last Chapter 7 concludes the work.

## 2 Background

Several low-power techniques such as dynamic voltage and frequency scaling (DVFS) [36], clock gating [47], quasi-adiabatic clocking [13], and LC resonant clocking [1, 2, 7, 15, 24, 40, 42] are commonly used to address high power consumption in clock networks. In a traditional clocking method, half of the switching power is utilized to charge a capacitive node when the clock transitions from 0-to-1. The other half of switching power is dissipated in the discharge cycle, when the clock transitions



**Fig. 2** Energy recycling techniques include (a) Quasi-adiabatic clocking topology places a capacitor in parallel with clock load to store energy [13], (b) Parallel resonance topology places an inductor in parallel with clock load [40], (c) Series resonance topology places an inductor in the discharge path of clock load [2], and (d) Quasi-resonant clocking topology places an inductor and an additional transistor to conditionally disconnect the inductor [42]

from 1-to-0. Energy recycling techniques harvest this dissipated energy and recycle it during the next clock transition.

## 2.1 Energy Recycling Techniques

Widely used energy recycling techniques include quasi-adiabatic clocking and LC resonant clocking, as shown in Fig. 2. Quasi-adiabatic clocking, as shown in Fig. 2a, uses a capacitor ( $C_{\text{adiabatic}}$ ) to store the dissipated energy from the load capacitance ( $C_{\text{load}}$ ). This  $C_{\text{adiabatic}}$  capacitor is placed in parallel to the load capacitance. When two capacitors are in parallel at different potentials, current will flow from the higher potential to the lower potential until the two capacitors have the same potential. At this point, no current flow occurs and all nodes are in steady state. If  $C_{\text{adiabatic}}$  is not disconnected during some portions of the cycle, the clock buffer will have to charge/discharge the total capacitive load  $C_{\text{load}} + C_{\text{adiabatic}}$  each cycle which would increase the overall power consumption to  $(C_{\text{load}} + C_{\text{adiabatic}}) V_{dd}^2 f$ . The quasi-adiabatic clocking employs a pass gate that would disconnect the  $C_{\text{adiabatic}}$  capacitor, depending upon control signals that determine the duration of the energy recovery and reuse states [13].

Among several energy recycling techniques, LC resonant is most widely used as it precisely replicates conventional complementary metal-oxide semiconductor (CMOS)

clocking and demonstrates great savings in dynamic power consumption [2] [19]. There are several LC resonant techniques, such as parallel resonance [40] (Fig. 2b), series resonance [2, 4] (Fig. 2c), intermittent resonance [15], and quasi-resonance [42] (Fig. 2d).

However, most LC resonance techniques suffer from a limitation of narrow frequency band and high skew. Parallel resonance topology, as shown in Fig. 2b, places an inductor in parallel to the clock load capacitance ( $C_{\text{load}}$ ) to store some of the dissipated energy. This technique saves the highest amount of power only when the system clock frequency equals the resonant frequency. On the other hand, quasi-resonance (Fig. 2d) employs an additional transistor and multiple control signals that conditionally disconnect the inductor twice every clock cycle. This increases the circuit complexity and the area of the clock driver.

To overcome the narrow frequency band, researchers utilize series resonance techniques [2]. This approach uses an inductor placed in the discharge path to store the dissipated energy in the form of a magnetic field. This energy is recycled in the next rising clock edge.

Figure 3a shows the equivalent LC resonant tank of the series resonant circuit which helps determine the resonant frequency and the inductor size. The resistance  $R_T$  represents the combined resistance of the NMOS transistor ( $R_r$ ), parasitic wiring resistance ( $r_w$ ), and the resistance of the inductor ( $r_L$ ). The series inductor allows the energy stored on the load capacitor to be transferred to the  $V_{\text{DD}}/2$  node and then recovered back immediately to make the output go high. This work uses a dynamic voltage divider circuit to generate this bias voltage.

Figure 3b shows the output at node  $V_C$ . The free energy swing obtained as a result of recycled dissipated energy, which is the difference between resonant high output ( $V_{\text{OH}}$ ) and resonant low output ( $V_{\text{OL}}$ ) [2], can be expressed as:

$$V_{\text{OH}} - V_{\text{OL}} = \frac{V_{\text{DD}}}{2}(1 + e^{-\pi/Q}) - \frac{V_{\text{DD}}}{2}(1 - e^{-\pi/2Q}) \quad (1)$$

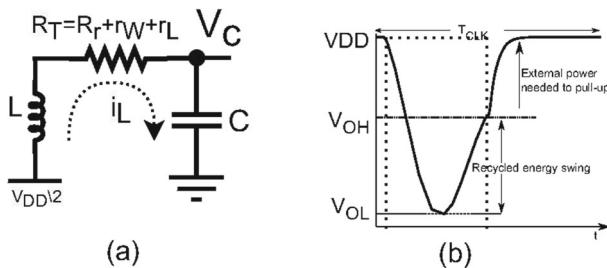
Here,  $Q$  is the quality factor of the inductor, which is given by  $Q = \sqrt{L/(CR^2)}$ . We would need an external power source to pull the output from  $V_{\text{OH}}$  to  $V_{\text{DD}}$  as shown in Fig. 3b. The resonant frequency  $f_{\text{RES}}$  at which the inductor resonates is shown in Eq. (2)

$$f_{\text{RES}} = \frac{1}{T_{\text{RES}}} = \frac{1}{2\pi\sqrt{LC}} \quad (2)$$

Here,  $T_{\text{RES}}$  is the resonant time period. A load capacitance of 2.5pF would require a 0.4nH inductor running at 5 GHz resonant frequency  $f_{\text{RES}}$ .

## 2.2 State-of-the-Art Low-Power Flip-Flops

Several prior works have focused on low-power flip-flop designs. In [45], an 18T flip-flop was designed to achieve 40% improvement in energy/cycle compared to conventional primary–secondary flip-flop (PSFF). However, to mitigate voltage degradation caused due to the non-complementary topology, they used a poly bias technique



**Fig. 3** When the input  $CLK$  goes “1” (in Fig. 2 (c)), **a** Equivalent series LC resonant tank formed from LC series resonance topology; **b** Output at capacitor node  $V_C$  shows the recycled energy swing from  $V_{OL}$  to  $V_{OH}$

[6]. This requires extra design effort. In [6], an 18T single-phase clocked flip-flop was designed for low-power operation. It showed 68% lower power consumption in overall power at 0.6V but had functionality limitations when the voltage was scaled, as reported in [43, 51]. In [22], a flip-flop with energy recovering clock using LC resonance was introduced, but it was specifically designed for soft-error hardening. In [12], a dual-edge triggered sense amplifier flip-flop was designed to achieve low power and area. However, this flip-flop uses unconventional dual-edge triggering which is not suitable for resonant operations. For reliability and robust operation, we implement resonant pulsed flip-flops based on widely used traditional pulsed registers [38] and resonant TSPC flip-flops based on traditional TSPC registers [27].

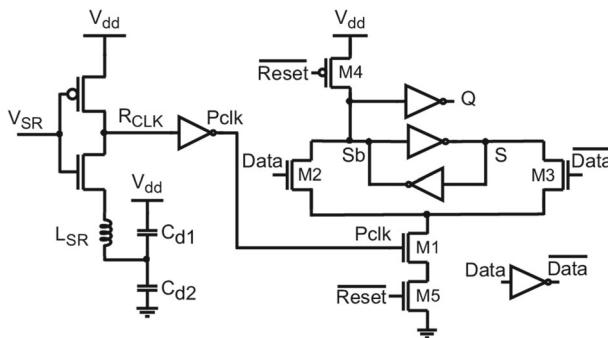
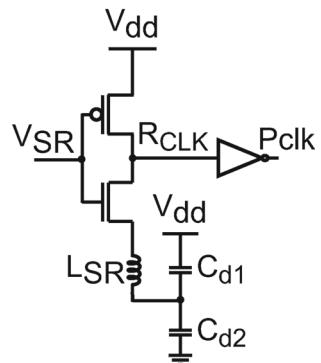
### 3 Proposed Wideband Resonant Flip-Flops

This section presents the proposed resonant flip-flops. The input clock signal to the flip-flops is a boosted-amplitude pulsed signal generated using a matching pulse generator (which will be discussed in detail in Sect. 4.1). The resonant flip-flops utilize on-chip inductors to recycle the clock power consumed by employing a pulsed series resonant driver.

#### 3.1 Pulsed Series Resonance Driver (PSR)

Figure 4 illustrates the pulsed series resonant driver (PSR). A boosted signal  $V_{SR}$ , generated using a matching pulse generator (shown in Fig. 8), is provided as the input to multiple PSR drivers to generate a pulse signal  $R_{CLK}$ . The inductors on the PSR drivers resonate with the capacitive load of the clock tree to generate a pulse signal that is traversed through many levels of clock gaters and clock buffers. Since we provide a boosted  $V_{SR}$  signal as the input, we obtain a rail-to-rail swing at the output of the PSR driver. Then, the output signal of the PSR driver  $R_{CLK}$  is inverted and supplied to pulsed registers as the clock input signal  $P_{clk}$ . The capacitors  $C_{d1}$  and  $C_{d2}$  form a dynamic voltage divider circuit to produce the bias voltage required for the series inductor to store the energy. The size of  $C_{d2}$  is  $4 \times C_{d1}$ . These capacitors are typically

**Fig. 4** Pulsed series resonant driver (PSR) generates a clock pulse using energy recycling series LC resonance. This clock pulse drives the register stage



**Fig. 5** 13T pulsed flip-flop (13TPFF) uses a PSR driver to recycle the input clock signal power

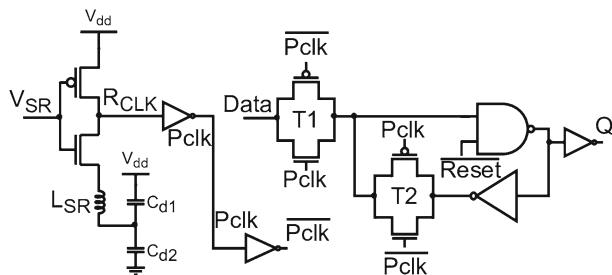
10× larger than the total load capacitance  $C_{\text{load}}$ . These capacitors are implemented once as on-chip lumped capacitors and shared between all the resonant drivers.

### 3.2 Resonant 13T Pulsed Flip-Flops

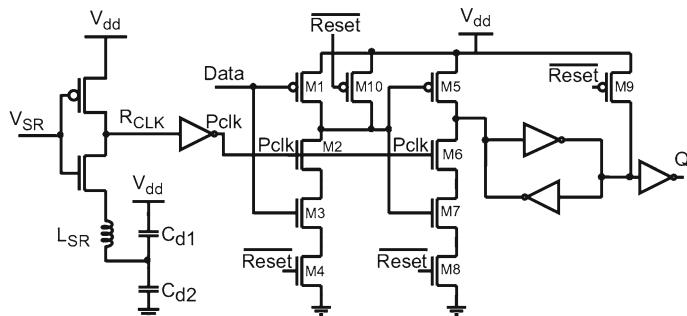
The transistor-level implementation of the proposed 13T pulsed flip-flop is shown in Fig. 5. It takes the input data and inverts it to provide it to the transistors  $M_2$  and  $M_3$ , respectively. The  $M_2$  and  $M_3$  transistors' drain is connected to the storage cells where the data are stored as logic “1” or a logic “0.” If a “1” is stored in the register, the value at  $S = “1”$  and  $S_B = “0.”$  If a “0” is stored, the voltages would be reversed.

When  $P_{\text{clk}}$  is “0,” the transistor  $M_1$  is turned off, wherein the FF is in hold/retain state, and the values of  $S$  and  $S_B$  are unaltered. Consider the case when  $\text{Data} = “1”$  and  $P_{\text{clk}} = “1,”$  the transistors  $M_2$  and  $M_1$  turn on connecting the node  $S_B$  to ground, which then discharges the node and makes it “0” making  $Q = “1,”$  thus writing a “1” into the register. When  $\text{Data} = “0”$  and  $P_{\text{clk}} = “1,”$  the transistors  $M_3$  and  $M_1$  turn “ON,” writing a “0” at node  $Q.$

The flip-flop has an active-low asynchronous reset signal. The  $M_4$  and  $M_5$  transistors are turned “ON” and “OFF,” respectively, when the  $\text{Reset}$  signal goes to low, resulting in a logic “1” at node  $S_B$  that writes a “0” to the output  $Q.$



**Fig. 6** Pulsed resonant flip-flop (PRFF) is based on a conventional pulsed register [38] and employs a PSR to drive the clock input pins of the register



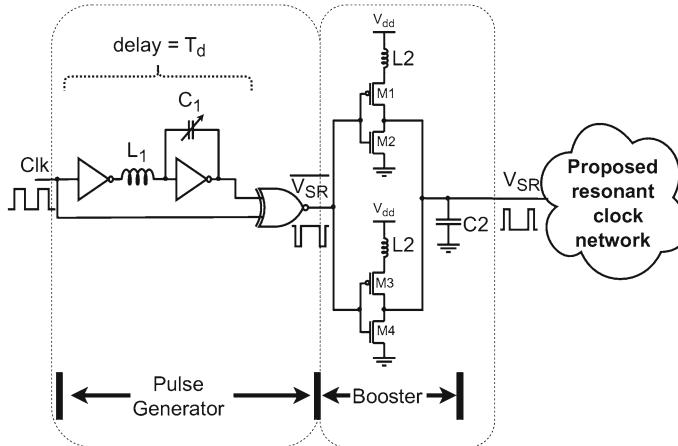
**Fig. 7** Resonant TSPC flip-flop (TSPCFF) is based on a TSPC register [38] and employs a PSR to generate pulse signal that drives the register

### 3.3 Pulsed Resonant Flip-Flop

Figure 6 depicts the pulsed resonant flip-flop (PRFF). When the  $P_{clk}$  is “0,” the flip-flop is in a hold/retain state. During this state, the transmission gate  $T2$  turns “ON,” and the feedback loop is open, retaining the previous data provided. When the  $P_{clk}$  = “1,” the transmission gate  $T1$  turns “ON,” copying the  $Data$  input to the  $Q$  output. The flip-flop has an active-low asynchronous  $Reset$  signal that sets the output  $Q$  to “0.”

### 3.4 Resonant TSPC Flip-Flop

Transistor-level implementation of the proposed resonant TSPC flip-flop (TSPCFF) is shown in Fig. 7. Similar to previous flip-flops, TSPCFF is also a positive edge triggered flip-flop that recycles the clock input provided using a PSR. The clock signal generated from PSR is provided to transistors  $M2$  and  $M6$ . When the  $P_{clk}$  signal is “0,” the transistors  $M2$  and  $M6$  are turned “OFF,” wherein the flip-flop is in hold/retain state as the input  $Data$  is disconnected from the storage cells. When  $P_{clk}$  is “1,” the transistors  $M2$  and  $M6$  turn “ON,” propagating the  $Data$  signal to the output  $Q$ . TSPCFF also has an active-low asynchronous  $Reset$  signal. When  $Reset$  goes low, transistors  $M9$  and  $M10$  turn “ON,” which sets the output  $Q$  to “0.”

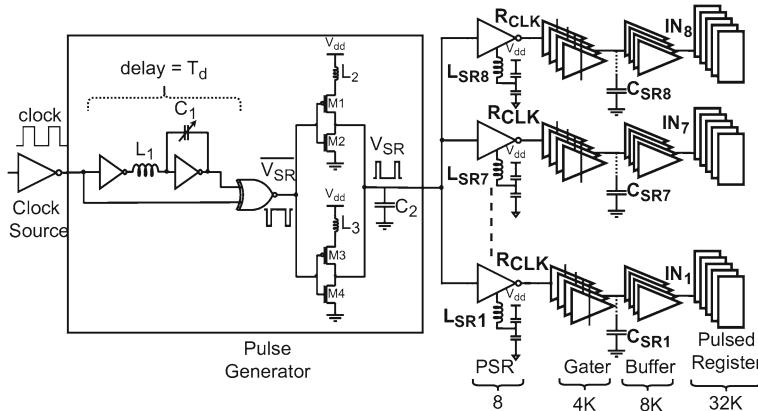


**Fig. 8** The matching pulse generator uses an XNOR gate to generate a pulse at both clock edges with pulse width  $T_d$ . The dual-rail booster generates a boosted-amplitude signal using series resonance by matching the shared inductor  $L_2$  with the load capacitance  $C_2$

## 4 Proposed Wideband Resonant Clock Networks

### 4.1 Matching Pulse Generator and Dual-Rail Booster

The series LC components are used to recover packets of energy wherever there is a large parasitic capacitance. However, series LC resonance requires the pulse width of the signal to match the LC value. The matching pulse generator circuit, depicted in Fig. 8, uses a matched LC-delay generation circuit to generate a pulse with the use of a programmable miller capacitor. This pulse is used to enable series resonance and does not recycle any energy. Tuning the LC-delay components to match the series LC of the clock network allows for the mitigation of variations caused by factors such as process, voltage, or temperature resulting in higher stability of the generated pulse width [3]. In the proposed architecture, the matching pulse generator uses the input from the system clock source and generates a pulse  $V_{SR}$  with boosted amplitude. The series inductor  $L_1$  and the matching miller capacitor  $C_1$  generate a delay of  $T_d = \pi\sqrt{L_1 C_1}$ . The clock and the delayed signal are fed into an XNOR gate to generate a pulse  $\overline{V_{SR}}$  at both clock edges with a pulse width  $T_d$ . Now, a voltage doubler circuit is employed to invert the generated dual-edge triggered pulse resulting in a boosted signal  $V_{SR}$ . The voltage doubler circuit uses the pulsed series resonance technique to generate a boosted signal. When the  $\overline{V_{SR}}$  is low, the PMOS transistors  $M_1$  and  $M_3$  are “ON,” and the inductor resonates with the load capacitance  $C_2$ , which represents the total capacitance load of multiple PSR drivers. For large load capacitances, the value of the series inductors is quite small. The inductor in the voltage doubler circuit can be adjusted according to the load of the matching pulse generator to produce a boosted signal  $V_{SR}$ . We use a dual-rail booster circuit to reduce the power consumed by the voltage doubler by decreasing the resistance of the pull-up network.



**Fig. 9** The proposed resonant clock tree architecture consists of a system clock source as the root, followed by a matching pulse generator, multiple PSR drivers utilizing on-chip inductors, clock gaters, clock buffers, and finally, various sets of resonant pulsed FF as leaf nodes

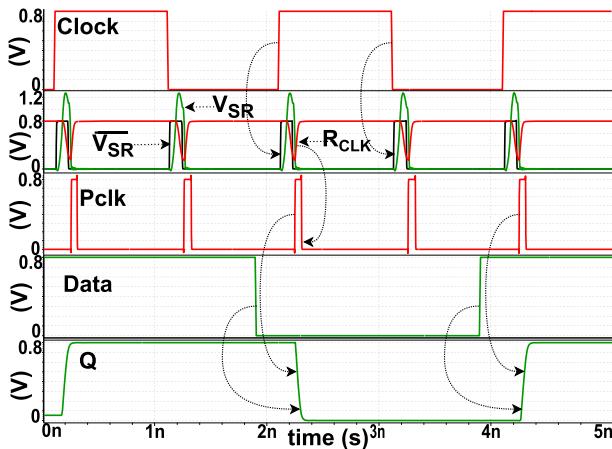
## 4.2 Resonant Clock Tree Architecture

The proposed resonant clock tree architecture utilizes a matching pulse generator, PSR drivers with on-chip inductors, clock gaters, clock buffers, and resonant flip-flops, as depicted in Fig. 9. The resonant clock tree network is designed based on the conventional clock tree network that distributes the input clock signals from a single clock root to all the elements. The functionality simulation for the resonant clock tree is shown in Fig. 10. The input clock source generates a 50% duty cycle clock that is converted into a pulse signal  $V_{SR}$  using the matching pulse generator (Fig. 8). This boosted signal is then propagated through multiple PSR drivers to recycle the power consumed in the network. The  $R_{CLK}$  signal generated is then propagated through multiple clock gaters and clock buffers to provide it as the input to the clock pins of resonant flip-flops.

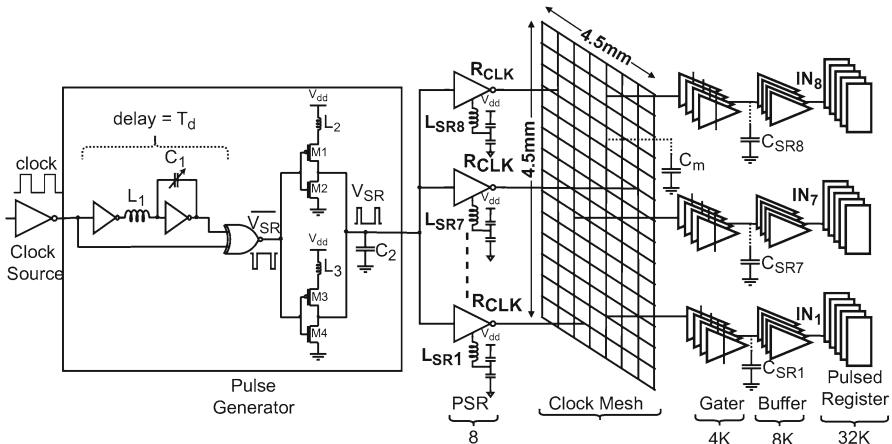
The inductance values of the on-chip inductors,  $L_{SR1}$ ,  $L_{SR7}$ , to  $L_{SR8}$  are matched with the total capacitance of the respective branch capacitances  $C_{SR1}$ ,  $C_{SR7}$ , to  $C_{SR8}$ . The inductor matching technique explains the methodology for calculating the inductance values described in Sect. 5.1.

## 4.3 Resonant Clock Mesh Architecture

High-performance processor designs utilize clock mesh architecture due to its high variation tolerance. A clock mesh is usually placed at the end of the clock driver network to distribute the clock signal to its load elements [39]. It is made using sets of vertical and horizontal metal wires. A clock mesh usually comprises parallel buffers driven by a clock tree from the top level [17]. It distributes the clock signal from the top-level tree making the clock signal accessible to all nodes of the mesh, resulting in a lower skew value. However, since the mesh consumes large amounts of metal, it has high switching capacitance resulting in increased power consumption.



**Fig. 10** Simulation waveforms show a CLK input of 0.5 GHz is provided to generate a 1 GHz *Pclk* clock of 1 GHz in order to assert as a clock input to the resonant flip-flops



**Fig. 11** The proposed wideband resonant clock mesh architecture is similar to resonant clock tree architecture, except the drivers are shorted, creating a clock mesh structure that is connected to the loads

Compared to a resonant clock tree architecture, in the case of resonant clock mesh, the inductor on the PSR driver is not matched with the local branch capacitance. Instead, the absolute capacitance of the mesh is matched with the inductor to recycle the power and minimize the skew generated by the clock mesh. This inductance is then distributed at various points on the mesh for the clock to be easily accessible at all the mesh nodes. Since the mesh consumes large amounts of metal, it has high switching capacitance resulting in increased power consumption. However, this high capacitance results in a low total inductor value.

The resonant clock mesh architecture, as shown in Fig. 11, places a clock mesh at the end of PSR drivers. Similar to resonant clock tree architecture, the PSR drivers receive a boosted input signal  $V_{SR}$ . In case of the clock mesh, the capacitance of the mesh

( $C_m$ ) and the branch capacitances ( $C_{SR1}, C_{SR7}, \dots, C_{SR8}$ ) form the load capacitance ( $C_{load}$ ) (in Equation. 4), which is in series with resonant inductors ( $L_{SR1}, L_{SR7}, \dots, L_{SR8}$ ).

## 5 Skew Reduction Methodology

Skew is defined as the temporal variation in the arrival time of clock transition at two different locations. There are several reasons for skew to occur, and one such cause is different loads on clock drivers [39]. This section explains the skew reduction methodology in clock network architectures using the proposed inductor tuning technique.

### 5.1 Inductor Tuning Technique

In a clock network, different clock tree branches have different capacitances caused by different loads and on-chip variations (OCV). This capacitance mismatch between different branches of a clock network will result in different clock arrival times. With the introduction of an inductor ( $L$ ), each branch of the clock tree represents a separate LC resonant tank. The inductance for each PSR driver can be determined using Equation. 4.

$$F_{res} = \frac{1}{2 \times DC_{rez}} = \frac{1}{2\pi\sqrt{L \times (C_{load})}} \quad (3)$$

$$L = \frac{DC_{rez}^2}{\pi^2 \times (C_{load})} \quad (4)$$

where  $DC_{rez}$  is defined by pulse width  $T_d$  generated by the matching pulse generator (Fig. 8) and  $C_{load}$  is the load capacitance of the inductor  $L$ .

The on-chip inductors  $L_{SR1}$  to  $L_{SR8}$  are matched with the load capacitances  $C_{SR1}$  to  $C_{SR8}$ , respectively, to have equal frequencies (Fig. 9). This inductor matching would result in equal frequency signals in all the clock branches, thus, reducing the skew. Wide band frequency of operation will not be affected, as the resonant frequency is independent of the input clock frequency. The primary reason is the  $delay = T_d$  of the matching pulse generator circuit being independent of input clock pulse width, and it works on the clock edges. Thus, for all the clock frequencies less than the derived resonant frequency  $f_{RES}$ , we can have the same inductor value that results in reduced skew. Moreover, skew can be minimized at run time by varying resonant pulse width  $T_d$ , to compensate for capacitance and inductance mismatch caused by OCV and/or process variations. This would mitigate the resulting skew by having a knob to vary the equivalent inductance with granularity. We have the ability to perform power-up calibration of this pulse width  $T_d$  using programmable registers.

## 5.2 Algorithm for Skew Reduction in Clock Tree Networks

The design methodology for placing and sizing the PSR drivers for an existing resonant clock tree architecture is described in Algorithm I. As input, the algorithm takes a clock network ( $clk_{tree}$ ), list of branch capacitances ( $C_{br}$ ), system clock frequency ( $F_{clk}$ ), duty cycle for resonance ( $DC_{rez}$ ), and a maximum skew constraint ( $S_{max}$ ) along with a predetermined range of inductor quality factor ( $Q_{range}$ ). The output of the algorithm is a resonant clock tree.

---

### Algorithm I. Determining the resonant drivers and inductor size for resonant tree

---

```

1: Input: Input network ( $clk_{tree}$ ), Branch capacitances ( $C_{br}$ ), System clock frequency ( $F_{clk}$ ), Duty cycle for resonance ( $DC_{rez}$ ), Skew constraint ( $S_{max}$ ), Q-factor range ( $Q_{range}$ );
2: Output: Resonant clock tree;
3:  $N_{Dr} = ReplaceDrivers(clk_{tree})$ ; ▷ replace the drivers in the network with PSR drivers
4:  $L_{Dr} = IndSizing(C_{br}, DC_{rez}, N_{Dr}, Q_{range})$ ; ▷ find L based on load capacitance of the individual branch
5: TransientSimulation()
6: while swing at all nodes  $\leq 90\%(VDD)$  do ▷ within a time period  $T_d$ 
7:    $driverSizeNew = ResizeDriver(driverSize)$ ; ▷ increase driver size
8:   if swing at all nodes  $\leq 90\%(VDD)$  then
9:      $N_{DrNew} = PartitionBranch(localbranch)$ ; ▷ partition the branch
10:     $PlaceDriver(N_{DrNew})$ ; ▷ place new PSR drivers
11:     $IndSizing(C_{br}, DC_{rez}, N_{DrNew}, Q_{range})$ ; ▷ calculate the inductor size for new PSR drivers
12:   end if
13: end while
14: TransientSimulation()
15:  $BranchList = Sort(N_{Dr}, latency)$ 
16: while skew  $> S_{max}$  do
17:    $N_{DrNew} = PartitionBranch(localbranch)$  ▷ partition the branch having highest latency
18:    $PlaceDriver(N_{DrNew})$ ; ▷ place new PSR drivers
19:    $IndSizing(C_{br}, DC_{rez}, N_{DrNew}, Q_{range})$  ▷ tune the inductors for new PSR drivers
20: end while

```

---

We begin with a conventional clock network and replace the drivers with PSR drivers using *ReplaceDrivers()* function as shown in Line 3. The inductance for each PSR driver can be determined by the function *IndSizing()* described in Line 4. The function *IndSizing()* calculates the inductor sizing using Equation 4. Detailed analysis of the calculated inductor values corresponding to its respective branch capacitances is shown in Fig. 16.

After the initial drivers are replaced, and the inductor size is determined, we run transient simulations on the network using Synopsys PrimeSim HSPICE to extract rise times, fall times, and initial skew. The while loop (Lines 6–13) checks for the output swing of the replaced PSR drivers. The output voltage swing of the PSR drivers is verified within a time period  $T_d$ . This time period  $T_d$  is calculated from the second rising edge of the clock to the falling edge. During this period, if the voltage swing of output is less than 90% (VDD), we increase the driver strength using the *ResizeDriver()* function, shown in Line 7. If the output does not have a VDD to  $V_{OL}$  swing, we partition the branch and place a new PSR driver for each branch. We also find the

proper inductor sizing for the new PSR drivers of the partitioned branches, as described between Lines 8–12.

The skew reduction methodology is described between Lines 16–20. We perform transient simulation again to get the latency of each branch and sort it using the *Sort()* function, as described in Line 15. The *Sort()* function sorts the latency of each resonant driver and arranges them in a list of descending order. The initial skew of the network is calculated from this list of latency values obtained. Between Lines 16–20, we check if the skew generated is higher than the input max skew constraint  $S_{max}$ . If it exceeds  $S_{max}$ , we partition the branch with the highest latency and place new PSR drivers (Lines 17–18) and also determine the new inductor sizing (Line 19). The algorithm exits when the skew obtained is within the max skew constraint  $S_{max}$ , or if there is no significant improvement in skew after an iteration. This methodology would result in a resonant clock network with reduced skew due to the inductor matching technique and lower power consumption as a result of resonant flip-flops used in the clock network.

### 5.3 Algorithm for Skew Reduction in Clock Mesh Networks

The design methodology of placing and sizing the PSR drivers for resonant mesh is described in Algorithm II. In this case, the algorithm initializes with a uniform clock mesh of  $(d \times d)$  dimension, mesh capacitance ( $C_m$ ), and load capacitances ( $C_l$ ) which are the capacitance values of the input uniform mesh and the load branches, respectively. It also takes in system clock frequency ( $F_{clk}$ ), duty cycle for resonance ( $DC_{rez}$ ), and a maximum skew constraint ( $S_{max}$ ) along with a predetermined range of inductor quality factor ( $Q_{range}$ ). The output of the algorithm is a resonant clock mesh network.

Depending upon the initial driver strength, we divide the mesh into smaller grids of size  $(m \times n)$  using *PartitionMesh()* (Line 3). A PSR driver is placed at the center of each partitioned grid using *PlaceDriver()* (Line 4). The inductance for each of the PSR driver is calculated using Eq. 4, which is defined using the function *IndSizing()* (Line 5). The load capacitance used to match the inductor value is determined by accumulating the local mesh capacitance ( $C_{mLocal}$ ) and its respective branch capacitances ( $C_{lLocal}$ ).

After the initial PSR drivers are placed, we run the transient simulation to verify if all the nodes of the clock mesh are having a  $V_{DD}$  to  $V_{OL}$  swing within the specified time period ( $T_d$ ). If the output voltage swing does not meet the required condition, the PSR drivers are resized using the *ResizeDriver()* function (Line 8), or the grid is split into two equal parts. We then place the PSR drivers and determine the new inductor values, shown between Lines 9–13. The algorithm iterates until the output swing at all nodes meets the specified criteria.

The skew reduction methodology is described between Lines 16–21. We perform another transient simulation to determine the latency of each branch and then sort the latencies obtained using the *Sort()* function (Line 16) in descending order. The initial skew of the clock mesh network is obtained from this list of latency values and is compared against the max skew constraint  $S_{max}$  provided. If the skew generated exceeds  $S_{max}$ , the local grid is divided into two equal grids, and then, the PSR driver

**Algorithm II.** Determining the resonant drivers and inductor size for resonant mesh

---

```

1: Input: Uniform mesh dimensions ( $d \times d$ ), Mesh capacitance ( $C_m$ ), Load capacitances ( $C_l$ ), System clock frequency ( $F_{clk}$ ), Duty cycle for resonance ( $DC_{rez}$ ), Skew constraint ( $S_{max}$ , Q-factor range ( $Q_{range}$ ));
2: Output: Resonant grid;
3:  $N_{Dr} = PartitionMesh(DM)$ ;  $\triangleright$  partition the grid into smaller  $m \times n$  grids;
4:  $PlaceDriver(N_{Dr})$ ;  $\triangleright$  place a driver at the center of each grid;
5:  $L_{Dr} = IndSizing(C_m, C_l, DC_{rez}, N_{Dr}, Q_{range})$ ;  $\triangleright$  find L based on load and mesh capacitance of the grid
6:  $TransientSimulation()$ 
7: while swing at all nodes  $\leq 90\%(VDD)$  do;  $\triangleright$  within a time period  $T_d$ 
8:    $driver_{sizeNew} = ResizeDriver(driver_{size})$ ;  $\triangleright$  increase driver strength
9:   if swing at all nodes  $\leq 90\%(VDD)$  then
10:     $N_{DrNew} = PartitionMesh(localmesh)$ ;  $\triangleright$  partition the grid
11:     $PlaceDriver(N_{DrNew})$ ;  $\triangleright$  place new PSR drivers
12:     $IndSizing(C_{mLocal}, C_{lLocal}, DC_{rez}, N_{DrNew}, Q_{range})$ ;  $\triangleright$  calculate the inductor size for new PSR drivers
13:   end if
14: end while
15:  $TransientSimulation()$ 
16:  $BranchList = Sort(N_{Dr}, latency)$ 
17: while skew  $> S_{max}$  do
18:    $N_{DrNew} = PartitionMesh(localmesh)$   $\triangleright$  partition the local grid having highest latency
19:    $PlaceDriver(N_{DrNew})$ ;  $\triangleright$  place new PSR drivers
20:    $IndSizing(C_{mLocal}, C_{lLocal}, DC_{rez}, N_{DrNew}, Q_{range})$   $\triangleright$  calculate the inductor size for new PSR drivers
21: end while

```

---

is placed at the center of each grid (Lines 18–20). This loop iterates until the skew generated is within the specified  $S_{max}$  or if there is no improvement of the skew generated. The final output would be a resonant clock mesh network with reduced skew due to the inductor matching technique and consumes lower power than a conventional clock mesh as a result of using resonant flip-flops.

## 6 Simulation Setup and Analysis of Power and Skew

### 6.1 Experimental Setup

The proposed resonant clock architectures were simulated using Synopsys PrimeSim HSPICE for ASAP 7 nm [9], and Synopsys SAED 14-nm fin-shaped field-effect transistor (FinFET) technologies [34]. We used accurate distributed RC models for clock distribution networks. We performed evaluations of the proposed resonant clocking architectures on two different clock distribution topologies, namely a conventional

clock tree architecture and conventional clock mesh architecture. Each network has eight clock drivers, 4K clock gaters, 8K clock buffers, and 32K FFs. The conventional clock network uses transmission gate primary–secondary FFs (PSFF), whereas the resonant clock network uses the proposed resonant FFs. Moreover, validation of the proposed skew reduction technique is demonstrated on industrial benchmarks, namely ISPD 2009 [26], ISPD 2010 [46], and ISCAS89 [5] testbench circuits for proposed clock networks. All the simulations are performed for frequencies ranging from 1 to 5 GHz.

## 6.2 Power and Delay Analysis of Resonant Flip-Flops

This section reports the power and delay analysis of the proposed resonant FFs, implemented in 14 nm technology. All the FFs layouts are compatible with a standard cell height of 24 horizontal M2 tracks. The normalized layout area, CLK-to-Q ( $t_{c-q}$ ) delay, setup times ( $ts$ ), hold times ( $th$ ), and power for the FFs are listed in Table 1. Among all the competing FFs, the 13TPFF consumed the highest layout area of  $9.62 \text{ }\mu\text{m}^2$ , which is  $1.75 \times$  the area of PSFF whose area is  $5.151 \text{ }\mu\text{m}^2$ , and  $2.9 \times$  the area of a PRFF whose area is  $3.091 \text{ }\mu\text{m}^2$ . The proposed 13TPFF has a  $ts$  of  $-25 \text{ ps}$  and a  $th$  of  $60 \text{ ps}$  with a clock-to-q delay of  $37.3 \text{ ps}$ . Empirically, pulsed register-based resonant FFs exhibit negative  $ts$ . Negative setup time means that the FF can latch the *Data* provided, even after the clock edge arrives, which tremendously impacts resolving setup time  $ts$  related timing issues. The PSFF has a  $ts$  of  $14 \text{ ps}$ ,  $th$  of  $2 \text{ ps}$ , and  $t_{c-q}$  of  $32.5 \text{ ps}$ . The resonant PRFF has a better  $ts$  of  $-95 \text{ ps}$  but has a high  $th$  of  $96 \text{ ps}$  which is similar to the TSPCFF with  $-92 \text{ ps}$   $ts$  and  $93 \text{ ps}$   $ts$ . However, the power consumed by the proposed 13TPFF is  $2 \times$  higher than the PSFF. Although the proposed resonant FFs consume higher power, they enable resonant clocking architecture that reduces the overall power consumption (which will be discussed in detail in Sect. 6.5). Among all the competing FFs, the PRFF consumes the lowest dynamic and static powers.

For measuring the performance and functionality of the proposed 13TPFF under process variations, we consider 5000 samples of CLK-to-Q ( $t_{c-q}$ ) delay using Monte Carlo simulation.  $\pm 10\%$  variation in the length of all devices is considered while performing the simulations. The  $t_{c-q}$  delay distributions of PSFF, PRFF, TSPCFF, and 13TPFFs are shown in Fig. 12a–d, respectively. Among all the resonant FFs, the PRFF has lowest mean  $t_{c-q}$  of  $35 \text{ ps}$  with standard deviation of  $0.066 \text{ ps}$ .

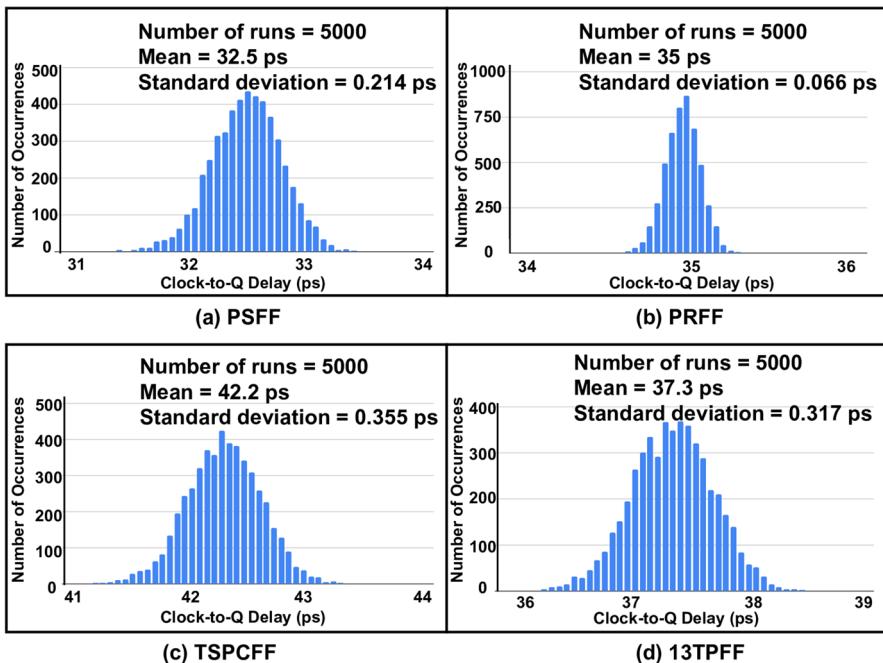
## 6.3 Power and Skew Analysis of Clock Tree Networks

The proposed resonant clock tree (Fig. 13b) using resonant FFs is compared with a conventional clock tree architecture (Fig. 13a) that uses primary–secondary FFs (PSFF). Figure 13 shows the testbench used for functional simulation implemented using standard 7-nm and 14-nm FinFET technology.

Power and skew comparisons of the proposed clock tree architecture for frequencies ranging from 1 to 5 GHz are shown in Table 2. The proposed clock tree architecture, while using 13TPFFs, saves 21.9% power in 14 nm technology, and 26.5% power while using 7 nm technology, compared to a conventional clock tree architecture

**Table 1** The proposed 13TPFF exhibits a better setup and hold times than the industry-standard PSFF while consuming more dynamic power and area; however, it consumes lower static power than PSFF and enables power saving in overall clock architecture

Type of Register	Normalized area	Delay (ps)			Static power (pW)		Dynamic Power ( $\mu$ W)				
		tc-q	ts	th	D = 0	D = 1	1 GHz	2 GHz	3 GHz	4 GHz	5 GHz
MSFF	1	32.5	14	2	1550	593	8.3	14.1	21	28	35.1
PRFF	0.59	35.1	– 95	96	278	272	7.16	13.8	20.4	27.1	33.8
TSPCFF	0.84	41.9	– 92	93	283	664	12.3	20.2	28	35.9	43.7
13TPFF	1.75	37.3	– 25	60	501	538	16.2	31.1	46	61	76

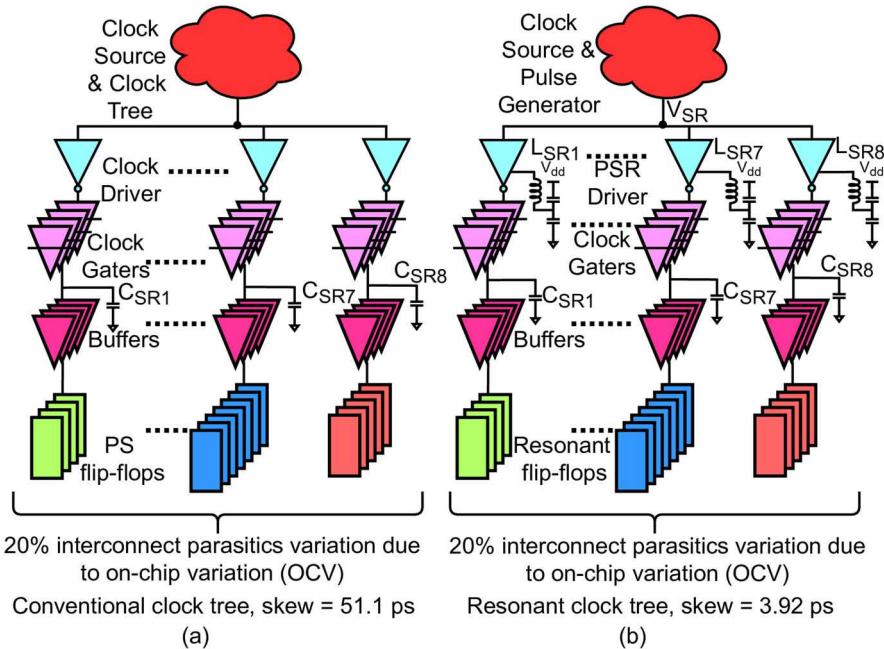


**Fig. 12** Illustration of Monte Carlo simulation results for various registers by considering 5000 samples with 10% length variation. In **a**, PS flip-flop has a mean of 32.5 ps with SD of 0.214 ps, in **b** PRFF has a mean of 35 ps with SD of 0.066 ps, in **c** TSPCFF has a mean of 42.2 ps with SD of 0.355 ps, and finally, in **d** our proposed 13TPFF has a mean of 32.3 ps with SD of 0.317 ps

with PSFFs. The skew generated by the traditional clock tree is 51.1 ps and 32.4 ps, using 14 nm and 7 nm, respectively. As a result of the inductor tuning technique (Sect. 5.1), the proposed resonant clock tree architecture reduces the skew generated by 92% and 87% in 14 nm and 7 nm technologies, respectively, while using 13TPFFs. Similarly, the proposed architecture reduces 27% power and 95% skew in 14 nm technology while using TSPCFFs, and 29% power and 88% skew in 7 nm technology. The proposed architecture saves highest power when using PRFFs. It saves 43% power while reducing the skew by 90% in 14 nm technology and saves 45.8% power and 87% skew using 7 nm technology.

#### 6.4 Power and Skew Analysis of Clock Mesh Networks

The proposed clock mesh architecture, as shown in Fig. 14, is also implemented using standard 7-nm and 14-nm FinFET technology. We use a distributed RC model to design the mesh. The clock mesh consists of  $32 \times 32$  grids totaling 4.5 mm  $\times$  4.5 mm in dimension. The load network used is the same as the previous testbench for resonant clock tree simulations (Fig. 13). Similarly, multiple simulations are performed for frequencies ranging from 1 to 5 GHz. Finally, the proposed resonant clock mesh



**Fig. 13** Clock tree architectures are used for functional simulations. **a** Conventional clock tree architectures consist of eight branches with different loads totaling 32k registers, and **b** resonant clock tree architecture replicates the same number of branches and loads as the conventional one

networks using resonant FFs are compared with a conventional clock mesh network using PSFFs.

The power and skew comparisons of clock mesh networks for frequencies ranging from 1 to 5 GHz are shown in Table 3. The proposed clock mesh architecture, while using 13TPFFs, reduces 27% and 26.9% of clock power over a conventional clock mesh architecture, in 14 nm and 7 nm technologies. The skew generated by the conventional clock tree is 108 ps and 74 ps, using 14 nm and 7 nm, respectively. As a result of the inductor tuning technique (Sect. 5.1), the proposed resonant clock mesh architecture reduces the skew generated by 91.5% and 89% in 14 nm and 7 nm technologies, respectively, while using 13TPFFs. Similarly, the proposed architecture reduces 31.9% power and 90.9% skew in 14 nm technology while using TSPCFFs and saves 33.2% power and 89.2% skew in 7 nm technology. The proposed clock mesh architecture saves the highest power when using PRFFs. It saves 44.6% power while reducing the skew by 90% in 14 nm technology and reduces power by 45.2% and skew by 88.6% in 7 nm technology.

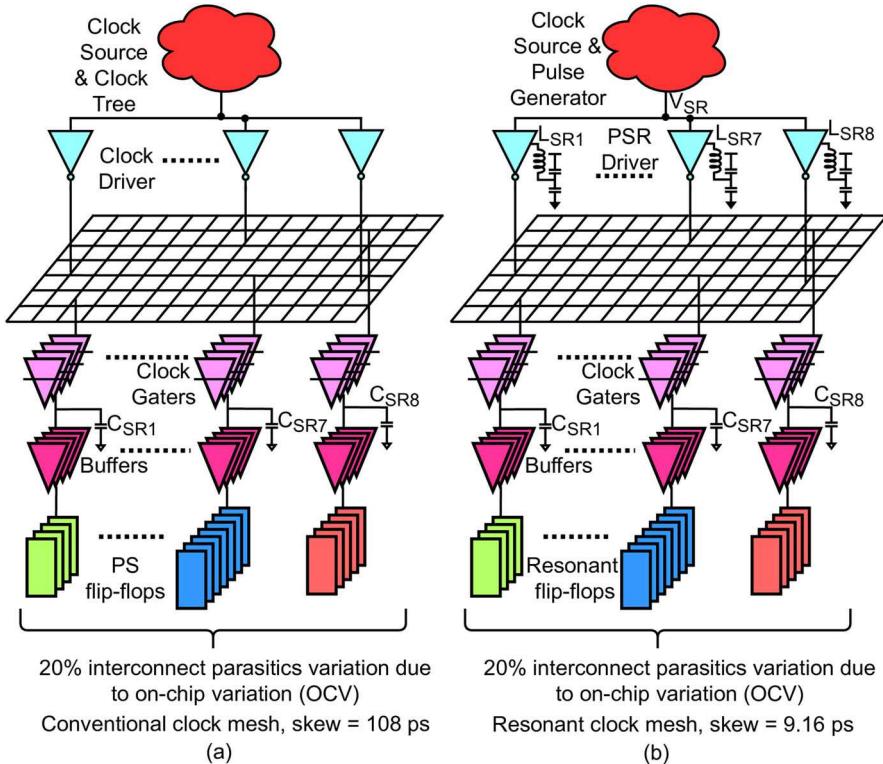
## 6.5 Power and Skew Analysis on Existing Standard Industrial Benchmarks

In order to validate the proposed skew reduction techniques in clock tree networks, we used ISPD 2009 testbench circuit (s1r1 with 81 sinks) [26], ISPD 2010 testbench

**Table 2** Analysis of resonant clock tree architectures using 14 nm and 7 nm technology nodes

Type of network	Technology node	Skew (ps)	Power consumed (mW)				
			1 GHz	2 GHz	3 GHz	4 GHz	5 GHz
PSFF network	14 nm	51.1	30.8	60.6	89.2	116	138
	7 nm	32.4	59.8	119	172	234	296
PRFF network	14 nm	4.61	17.4	34.1	50.3	65.7	78.7
	7 nm	3.95	32.5	64.9	92.4	127	160
TSPCFF network	14 nm	2.05	22.2	43.6	64.6	84.4	102
	7 nm	3.66	42.1	84.3	123	167	209
13TPFF network	14 nm	3.92	23.8	46.8	69.4	90.7	110
	7 nm	4.14	43.9	87.2	126	173	218

The power and skew values of flip-flop networks by scaling frequency from 1 to 5 GHz depict consistent power savings and skew reduction while using the proposed resonance technique



**Fig. 14** Clock mesh architectures are used for functional simulations. **a** Conventional clock mesh architectures with shorted drivers connecting loads, and **b** resonant clock mesh architecture using PSR drivers to drive the same load as the conventional one

circuit (01.in with 1107 sinks) [46], and ISCAS89 testbench circuit (s5378 with 179 sinks) [5]. Table 4 shows the comparisons of power and skew of the resonant networks.

While using the ISPD 2009 testbench, extracted from IBM ASIC design with 81 flip-flops, the conventional clock architecture has a skew of 29 ps while consuming 6.65 mW power at 1 GHz frequency. On the other hand, the resonant clock network with 81 PRFFs reduces the power consumption by 66% with a skew reduction of 85%. Moreover, the resonant clock network using TSPCFF saves 64% power while lowering the skew by 92%, and the resonant clock network using 13TPFF saves 59% power with 88% skew reduction.

The ISPD 2010 testbench circuit has more sink density with 1107 sinks. The conventional clock network using PSFF has a power consumption of 53.3 mW with 22.7 ps clock skew at 1 GHz frequency. The resonant clock architecture using PRFF also has 65% reduced power consumption with 83% reduced skew. Also, the resonant clock network with TSPCFF saves 61% power and 91% skew, and the resonant clock network using 13TPFF reduces 57% power consumption with 89.5% skew reduction.

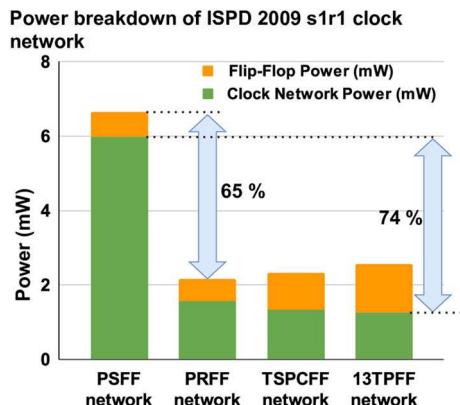
The ISCAS89 testbench circuit has 179 clock sinks and 2779 logic gates. The conventional clock architecture using PSFF generates 24.9 ps clock skew while consuming

**Table 3** The analysis of the proposed resonant technique on clock mesh architectures using 14 nm and 7 nm technology nodes

Type of network	Technology node	Skew (ps)	Power consumed (mW)				
			1 GHz	2 GHz	3 GHz	4 GHz	5 GHz
PSFF network	14 nm	108	35.3	68.5	102	131	162
	7 nm	74	74.4	151	221	294	368
PRFF network	14 nm	10.3	19.2	38.1	56.5	74.2	88.7
	7 nm	8.4	40.2	82.7	121	163	202
TSPCFF network	14 nm	9.74	23.5	46.7	69.3	91.1	110
	7 nm	7.94	49.8	102	149	195	242
13TPFF network	14 nm	9.16	25.2	50.1	74.4	97.9	118
	7 nm	8.12	54	110	164	215	267

The results obtained by scaling the frequency from 1 to 5 GHz support the resonant clock tree analysis showing similar power saving and reduction in skew even on a lower technology node

**Fig. 15** The breakdown of total power in ISPD 2009 testbench circuit (s1r1 with 81 sinks) depicts 74% lower power consumption in resonant clock architecture using 13TPFF compared to a conventional clock architecture using PSFF at 1 GHz, while the flip-flop power in 13TPFF clock architecture is 48% higher than PSFF clock architecture



14.1 mW power at 1 GHz frequency. On the other hand, the resonant clock architecture using PRFF saves 63% of power while reducing the skew by 87%. Moreover, the resonant clock network using TSPCFF has 61.7% reduced power consumption with 90% reduced skew, and the resonant clock network using 13TPFF has 57.6% reduced power consumption with 85% reduced skew.

Figure 15 illustrates the total power distribution of ISPD 2009 testbench circuit (s1r1 with 81 sinks) [26] in the proposed resonant clock architectures and conventional clock architecture using PSFF for 100% data activity rate at 1 GHz clock frequency. Although the proposed resonant flip-flops consume more power than PSFF, the resonant clocking architectures exhibit a lower total power consumption due to the energy recycled in the clock network. The proposed resonant clock architecture with 13TPFF consumes 74% less power in the clock network compared to conventional clock architecture with PSFF, while consuming 48% higher flip-flop power resulting in 57% reduced total power consumption. In addition, the clock network power in resonant clock architectures with TSPCFF has 72% lower clock power consumption and PRFF has

**Table 4** The power and skew analysis of the proposed resonant technique on clock tree architectures result in consistent power savings and balanced skew on the ISPD 2009 s1r1, ISPD 2010 01.in, and ISCAS89 s5738 circuits; the proposed PRFF network saves 64% power on average with 85% reduced skew at 1 GHz frequency

Benchmark	Type of network	Frequency (GHz)									
		1 GHz		2 GHz		3 GHz		4 GHz		5 GHz	
		Skew (ps)	Power (mW)	Skew (ps)	Power (mW)	Skew (ps)	Power (mW)	Skew (ps)	Power (mW)	Skew (ps)	Power (mW)
ISPD 2009 s1r1	PSFF network	29	6.65	33	12.7	31	18.62	37	25.4	34	30.25
	PRFF network	4.4	2.15	2.7	4.3	3.6	6.44	3.9	8.58	3.3	10.3
	TSPCFF network	2.28	2.32	2.4	4.55	2.3	6.79	2.9	9.02	2.4	10.8
	13TPFF network	3.72	2.57	4.3	5.14	4.2	7.6	3.8	10.1	4.1	12.2
ISPD 2010 01.in	PSFF network	22.7	53.3	26.7	107	33.4	160	31.3	213	42.6	269
	PRFF network	4.25	18.4	6.37	35.7	6.65	54.3	5.2	70.9	5.73	88
	TSPCFF network	2.1	20.3	3.32	39.2	2.92	59	4.22	75	3.7	92
	13TPFF network	2.4	23.4	3.4	44.7	3.1	67.4	3.5	89.2	4.1	105
ISCAS89 s5378	PSFF network	24.9	14.1	25.8	28.1	24.9	42.1	23.4	56.1	21.4	70.1
	PRFF network	3.11	5.09	3.72	9.94	3.16	14.4	3.94	19.2	3.31	22.9
	TSPCFF network	2.42	5.4	2.75	10.6	2.51	15.9	2.77	21.5	3.1	24.4
	13TPFF network	3.7	5.97	4.22	11.4	3.73	17.1	4.3	22.6	4.74	26.3

70% lower clock power consumption compared to a conventional clock architecture using PSFF.

The energy consumption of the proposed series resonant clocking architectures across 1–5 GHz frequency is shown in Table 5. The framework used to estimate the energy consumption for resonant clocking architecture is shown in Fig. 9. While using ISPD 2009 testbench circuit (s1r1 with 81 sinks), the resonant clock architecture using PRFF reduces the energy consumption by 66.2% on average compared to conventional clock architecture using PSFF across 1–5 GHz frequency, while resonant clock architecture using 13TPFF saves 59.9% energy on average. While using the ISPD 2010 testbench circuit (01.in with 1107 sinks), the resonant clock architecture using TSPCFF saves 63.7% energy on average across 1–5 GHz frequency, while the resonant architecture using PRFF saves 66.4% energy compared to conventional clock architectures using PSFF. The resonant clock architecture using PRFF saves 65.4% average energy consumption on ISCAS89 testbench circuit (s5378 with 179 sinks) while saving 59.7% average energy while using 13TPFF across 1–5 GHz clock frequency.

The heat generated throughout the circuit can be estimated through the junction temperature ( $T_j$ ) using the transient thermal resistance ( $\theta_{JA}$ ) for a pulse length  $t_p$ , which can be represented as:

$$T_j = T_A + P \times \theta_{JA}(t_p) \quad (5)$$

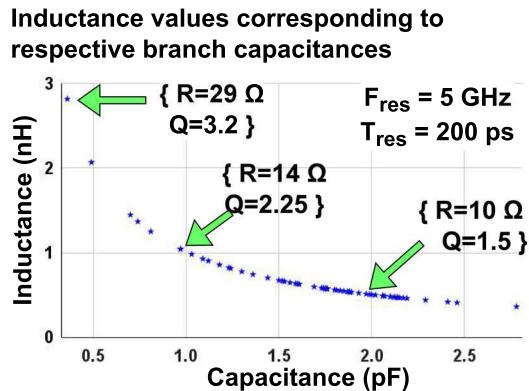
where  $P$  is the power consumed during  $t_p$ , and  $T_A$  is the ambient temperature [10]. Table 5 compares the junction temperatures of the proposed resonant clocking architectures with conventional clock architecture using PSFF at 5 GHz frequency with 27 °C ambient temperature ( $T_A$ ). The proposed resonant clock networks use lower power than conventional clock distribution networks and thus produce lower heat than existing clock networks. For example, a 48-pin Ceramic Leadless Chip Carrier (CLCC) package has a thermal resistance ( $\Theta_{JA}$ ) of 40 °C/W [33]. Considering a power difference of 0.181 W would result in a 7.2 °C higher temperature than the proposed resonant clock network for ISPD 2010 01.in circuit at 5 GHz frequency. For a typical SOC implementation with 10× the sinks as ISPD 2010 01.in circuit, a power difference of 1.81 W would result in 72.4 °C higher temperature than the proposed resonant clock network.

In a clock network, the variations in branch capacitances require different inductance values. Figure 16 plots the inductance and capacitance values extracted from the benchmark circuits. The resistance values are the sum of the resistance of inductor, wiring resistance, and n-transistor “ON” resistance. The quality factor ( $Q$ ) of the inductor is directly proportional to the inductance value and inversely proportional to capacitance and resistance values. Thus, larger capacitances result in low-quality factor, whereas higher inductance values result in large resistance values leading to a low-quality factor. Hence, while designing the inductors, we provide a range of quality factor that would result in higher power savings. A range of  $Q$  between  $\pi$  and  $\pi/3$  would result in higher power savings. Dedicated top two metal layers are used for implementing the inductors, which do not take any active silicon area [19]. The inductors can also be placed in dark silicon regions to achieve a higher quality factor.

**Table 5** The proposed series resonant clock architectures have 67.2% lower energy and 7.2°C lower junction temperature while using PRFF for ISPD 2010 01.in circuit compared to conventional clock architectures at 5 GHz frequency; besides the proposed 13TPFF has a 59.3% lower average energy consumption compared to conventional clock architectures using PSFF

Benchmark	Type of network	1 GHz		2 GHz		3 GHz		4 GHz		5 GHz	
		Energy (pJ)	Temp. (°C)								
ISPD 2009 s1r1	PSFF network	6.65	27.266	6.35	27.508	6.21	27.7448	6.35	28.016	6.05	28.21
	PRFF network	2.15	27.086	2.15	27.172	2.15	27.2576	2.15	27.3432	2.06	27.412
	TSPCFF network	2.32	27.0928	2.28	27.182	2.26	27.2716	2.26	27.3608	2.16	27.432
	13TPFF network	2.57	27.1028	2.57	27.2056	2.53	27.304	2.53	27.404	2.44	27.488
ISPD 2010 01.in	PSFF network	53.30	29.132	53.50	31.28	53.33	33.4	53.25	35.52	53.80	37.76
	PRFF network	18.40	27.736	17.85	28.428	18.10	29.172	17.73	29.836	17.60	30.52
	TSPCFF network	20.30	27.812	19.60	28.568	19.67	29.36	18.75	30	18.40	30.68
	13TPFF network	23.40	27.936	22.35	28.788	22.47	29.696	22.30	30.568	21.00	31.2
ISCAS89 s5378	PSFF network	14.10	27.564	14.05	28.124	14.03	28.684	14.03	29.244	14.02	29.804
	PRFF network	5.09	27.2036	4.97	27.3976	4.80	27.576	4.80	27.768	4.58	27.916
	TSPCFF network	5.40	27.216	5.30	27.424	5.30	27.636	5.38	27.86	4.88	27.976
	13TPFF network	5.97	27.2388	5.70	27.456	5.70	27.684	5.65	27.904	5.26	28.052

**Fig. 16** Inductance and capacitance values extracted from the benchmark circuits depict a smaller inductance value required for higher branch capacitances



**Table 6** A comparison of the proposed PRFF network with previous low-power clocking techniques depicts an average skew reduction of 78% in ISPD 2009 s1r1 benchmark and 80.8% while using ISPD 2010 01.in benchmarks

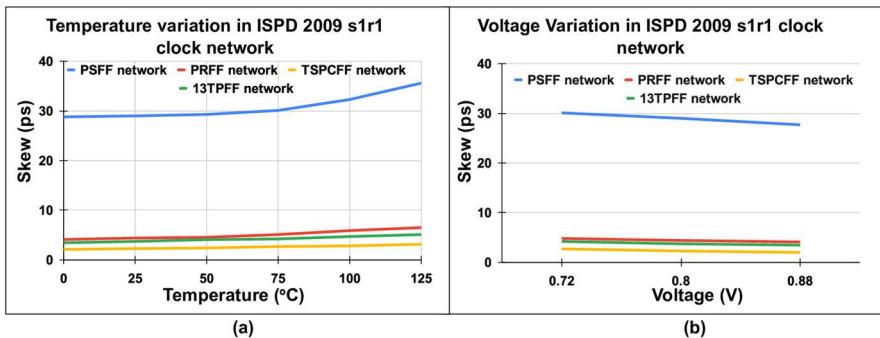
Benchmark	Methodology	Technology node	Frequency	Power (mW)	Skew (ps)
ISPD 2009 s1r1	ROCKS [17]	45 nm	1 GHz	71.1	20
	LARCS [18]	45 nm	1 GHz	–	–
	HCDN [21]	45 nm	1 GHz	20.2	21
	PRFF network	14 nm	1 GHz	2.15	4.4
ISPD 2010 01.in	ROCKS [17]	45 nm	1 GHz	179.3	77
	LARCS [18]	45 nm	1 GHz	368	32
	HCDN [21]	45 nm	1 GHz	38.2	11
	PRFF network	14 nm	1 GHz	18.4	4.25

An earlier implementation of resonant architecture in [19] showed a 2% area penalty when using the top two metal layers to implement the inductors.

Table 6 compares our results with implementation of distributed LC resonant clock grid synthesis (ROCKS) [17], library-aware resonant clock synthesis (LARCS) [18], and hybrid-mode clock distribution networks (HCDN) [21]. The ROCKS and LARCS are prior works on resonant grid synthesis. The HCDN clocking scheme uses global bufferless current-mode (CM) clocking and locally buffered voltage-mode (VM) clocking.

## 6.6 Effect of Temperature Variation on Skew

At lower technology nodes, the temperature variation results in major degradation of the performance of a chip. We used the ISPD 2009 testbench circuit (s1r1 with 81 sinks) [26] to quantify the temperature-variation-induced clock skew in a resonant network compared with a conventional clock network. Figure 17a shows the clock skew comparison at 1 GHz frequency with varying temperature from 0°C to 125°C. The conventional clock network using PSFF has a skew varying between 28.8 and 35.6 ps,



**Fig. 17** Effect of temperature and voltage variation on clock skew using ISPD 2009 testbench circuit shows that conventional clock architecture has 7 ps change in clock skew, whereas resonant clock architectures have 2 ps change. Also, while varying supply voltage by  $\pm 10\%$ , clock skew in conventional clock architectures varies by 3 ps, while resonant clock architectures have 1 ps clock skew variation

while the resonant clock architecture with PRFF has a skew variation between 4.1 and 6.5 ps. The lesser impact on resonant architecture is due to the reduced number of buffers compared to the conventional networks. Moreover, resonant clock architectures with TSPCFF have a skew variation between 2.1 and 3.15 ps and 13TPFF has a skew variation between 3.45 and 5.1 ps.

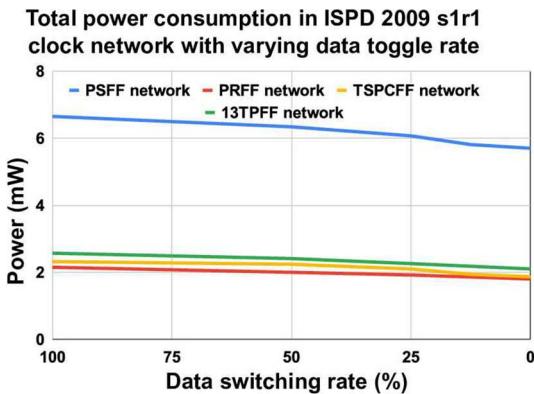
## 6.7 Effect of Supply Voltage Variation on Skew

Supply voltage variation is one of the major sources of variation in high-performance microprocessors. We use the same ISPD 2009 testbench circuit (s1r1 with 81 sinks) [26] to measure the clock skew variation induced by supply voltage variation considering the IR drops and localized rail voltage noise. Figure 17b shows the clock skew comparison at 1 GHz. We consider  $\pm 10\%$  variation in the supply voltage ( $V_{DD}$ ) from nominal  $V_{DD}$ . Conventional clock architecture with PSFF has a skew variation of 27.2–30.1 ps, while resonant clock architecture with TSPCFF has a skew variation of 1.99–2.7 ps. Moreover, resonant clock architecture has a skew variation of 4.1–4.8 ps while using PRFF and has a skew variation of 3.46–4.2 ps while using 13TPFF.

## 6.8 Effect of Data Switching Rate on Power Consumption

We measure the total power consumption of the ISPD 2009 testbench circuit (s1r1 with 81 sinks) by changing the data switching rate from 100 to 0% at 1 GHz frequency. A 100% data switching rate means, with each rising edge of the clock signal, the flip-flops will latch a new data at the output. Figure 18 compares the total power consumed in ISPD 2009 testbench circuit (s1r1 with 81 sinks) with varying data switching rate. We can observe that the resonant clock networks have higher power savings with high data switching rate, as the flip-flops toggle the data with every clock cycle. The proposed PRFF network saves 66% with 100% data switching rate, while it saves 64% with a 0% data switching rate.

**Fig. 18** Total power consumption in ISPD 2009 s1r1 clock with varying data toggle rates shows a higher power savings with high toggle rate



## 7 Conclusion

Power consumption and skew are major bottlenecks in high-performance microprocessor designs. This paper presents resonant clock network architectures to balance the skew and recycle the power consumed by employing resonant flip-flops and inductor tuning techniques. The proposed resonant flip-flops enable the clock networks to recycle the dissipated energy by placing an inductor in the discharge path. This inductor stores the dissipated energy in the form of a magnetic field to recycle it in the next rising clock edge. The proposed resonant clock tree network with 13TPFF saves 22% power with 91% skew reduction.

Furthermore, it saves 43.4% power with a 91% skew reduction while using PRFF in 14 nm technology, compared to a conventional PSFF-based clock tree architecture. Moreover, in 7 nm technology, resonant clock tree architecture with PRFF saves 45.8% power with 87.8% skew reduction. The proposed clock mesh network with 13TPFF saves 44.6% power with 90.4% reduced skew in 14 nm technology while saving 45.2% power and 87.8% skew reduction in 7 nm technology.

**Acknowledgements** This work was supported in part by Rezonent Inc. under Grant CORP-0061, National Science Foundation (NSF) award number: 2138253, and UMBC Startup grant. The authors also acknowledge M. Galib from UMBC for providing layouts data used in the analysis.

**Availability of Data and Materials** Data can be provided by the corresponding author upon reasonable request.

## Declarations

**Conflict of interest** The authors have no competing interests to declare.

## References

1. I. Bezzam, S. Krishnan, C. Mathiazhagan, T. Raja, F. Maloberti, Wide operating frequency resonant clock and data circuits for switching power reductions. *Analog. Integr. Circ. Sig. Process.* **82**, 113–124 (2015). <https://doi.org/10.1007/s10470-014-0447-1>

2. I. Bezzam, C. Mathiazhagan, T. Raja, S. Krishnan, An energy-recovering reconfigurable series resonant clocking scheme for wide frequency operation. *Trans. Circ. Syst. I.* **62**(7), 1766–1775 (2015). <https://doi.org/10.1109/TCSI.2015.2423797>
3. I. Bezzam, Reduced-power electronic circuits with wide-band energy recovery using non-interfering topologies. (2019). <https://patents.google.com/patent/US10340895B2>
4. I. Bezzam, Rawat Neelam, Digital circuits for radically reduced power and improved timing performance on advanced semiconductor manufacturing processes. (2021). <https://patents.google.com/patent/US11073861B2>
5. F. Brglez, D. Bryan, K. Kozminski, Combinational profiles of sequential benchmark circuits, in *International Symposium on Circuits and Systems (ISCAS)*, (1989), pp. 1929–1934 <https://doi.org/10.1109/ISCAS.1989.100747>
6. Y. Cai, A. Savanth, P. Prabhat, J. Myers, A. Weddell, T. Kazmierski, Ultra-low power 18-transistor fully static contention-free single-phase clocked flip-flop in 65-nm CMOS. *J. Solid-State Circ.* **54**(2), 550–559 (2019). <https://doi.org/10.1109/JSSC.2018.2875089>
7. D. Challagundla, M. Galib, I. Bezzam, R. Islam, Power and skew reduction using resonant energy recycling in 14-nm FinFET clocks, in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, (2022), pp. 268–272 <https://doi.org/10.1109/ISCAS48785.2022.9937771>
8. L. Cherif, M. Chentouf, J. Benallal, M. Darmi, R. Elgouri, N. Hmina, Usage and impact of multi-bit flip-flops low power methodology on physical implementation, in *2018 4th International Conference on Optimization and Applications (ICOA)*, (2018), pp. 1–5 <https://doi.org/10.1109/ICOA.2018.8370498>
9. L.T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, G. Yeric, ASAP7: a 7-nm finFET predictive process design kit. *Microelectron. J.* **53**(7), 105–115 (2016). <https://doi.org/10.1016/j.mejo.2016.04.006>
10. D. Edwards, H. Nguyen, Semiconductor and IC Package Thermal Metrics (rev. C), Texas Instruments. <https://www.ti.com/lit/an/spra953c/spra953c.pdf>
11. W.M. Elsharkasy, A. Khajeh, A.M. Eltawil, F.J. Kurdahi, Reliability enhancement of low-power sequential circuits using reconfigurable pulsed latches. *Trans. Circ. Syst. I.* **64**(7), 1803–1814 (2017). <https://doi.org/10.1109/TCSI.2017.2680433>
12. S.E. Esmaeili, R. Islam, A.J. Al-Khalili, G.E.R. Cowan, Dual-edge triggered sense amplifier flip-flop utilizing an improved scheme to reduce area, power, and complexity, in *19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, (2012), pp. 292–295 <https://doi.org/10.1109/ICECS.2012.6463565>
13. H.A. Fahmy, P-Y. Lin, R. Islam, M.R. Guthaus, Switched capacitor quasi-adiabatic clocks, in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, (2015), pp. 1398–1401 <https://doi.org/10.1109/ISCAS.2015.7168904>
14. T. Fischer, S. Arekapudi, E. Busta, C. Dietz, M. Golden, S. Hilker, A. Horiuchi, K.A. Hurd, D. Johnson, H. McIntyre, S. Naffziger, J. Vinh, J. White, K. Wilcox, Design solutions for the Bulldozer 32nm SOI 2-core processor module in an 8-core CPU. In *International Solid-State Circuits Conference*, (2011), pp. 78–80 <https://doi.org/10.1109/ISSCC.2011.5746227>
15. H. Fuketa, M. Nomura, M. Takamiya, T. Sakurai, Intermittent resonant clocking enabling power reduction at any clock frequency for near/sub-threshold logic circuits. *J. Solid-State Circ.* **49**(2), 536–544 (2014). <https://doi.org/10.1109/JSSC.2013.2294172>
16. J.L. Hennessy, D.A. Patterson, *A New Golden Age for Computer Architecture*. (2019), pp. 48–60 <https://doi.org/10.1145/3282307>
17. X. Hu, M.R. Guthaus, Distributed LC resonant clock grid synthesis. *Trans. Circ. Syst. I.* **59**(11), 2749–2760 (2012). <https://doi.org/10.1109/TCSI.2012.2190671>
18. X. Hu, W. Conley, M.R. Guthaus, Library-Aware Resonant Clock Synthesis (LARCS), in *Proceedings of the 49th Annual Design Automation Conference*, (2012), pp. 145–150 <https://doi.org/10.1145/2228360.2228389>
19. R. Islam, B. Saha, I. Bezzam, Resonant energy recycling SRAM architecture. *Trans. Circ. Syst. II.* **68**(4), 1383–1387 (2021). <https://doi.org/10.1109/TCSI.2020.3029203>
20. R. Islam, M.R. Guthaus, CMCS: current-mode clock synthesis. *Trans. Very Large Scale Integr. (VLSI) Syst.* **25**(3), 1054–1062 (2017). <https://doi.org/10.1109/TVLSI.2016.2605580>
21. R. Islam, M.R. Guthaus, HCDN: hybrid-mode clock distribution networks. *Trans. Circ. Syst. I.* **66**(1), 251–262 (2019). <https://doi.org/10.1109/TCSI.2018.2866224>
22. R. Islam, Low-power resonant clocking using soft error robust energy recovery flip-flops. *J. Electron. Test.* **34**, 471–485 (2018). <https://doi.org/10.1007/s10836-018-5737-6>

23. R. Islam, H.A. Fahmy, P.Y. Lin, M.R. Guthaus, DCMCS: highly robust low-power differential current-mode clocking and synthesis. *Trans. Very Large Scale Integr. VLSI Syst.* **26**(10), 2108–2117 (2018). <https://doi.org/10.1109/TVLSI.2018.2837681>

24. R. Islam, *High-Speed Energy-Efficient Soft Error Tolerant Flip-flops* (2011). <https://spectrum.library.concordia.ca/id/eprint/15130/>

25. R. Islam, H.A. Fahmy, P.Y. Lin, M.R. Guthaus, Differential current-mode clock distribution, in *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, (2015), pp. 1–4 <https://doi.org/10.1109/MWSCAS.2015.7282042>

26. ISPD-2009, *Proceedings of the 2009 International Symposium on Physical Design*, (2009). <https://www.ispd.cc/contests/09/ispd09cts.html>

27. S.M. Jahinuzzaman, R. Islam, TSPC-DICE: A single phase clock high performance SEU hardened flip-flop, in *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, (2010), pp. 73–76 <https://doi.org/10.1109/MWSCAS.2010.5548564>

28. H. Jeong, T.W. Oh, S.C. Song, S.O. Jung, Sense-amplifier-based flip-flop with transition completion detection for low-voltage operation. *Trans. Very Large Scale Integr. (VLSI) Syst.* **26**(4), 609–620 (2018). <https://doi.org/10.1109/TVLSI.2017.2777788>

29. A.A. Khan, A. Ali, M. Zakarya, R. Khan, M. Khan, I.U. Rahman, M.A.A. Rahman, A migration aware scheduling technique for real-time aperiodic tasks over multiprocessor systems. *IEEE Access*. **7**, 27859–27873 (2019). <https://doi.org/10.1109/ACCESS.2019.2901411>

30. N. Kumar, D.P. Vidyarthi, A novel energy-efficient scheduling model for multi-core systems. *Clust. Comput.* **24**, 643–666 (2021). <https://doi.org/10.1007/s10586-020-03143-w>

31. S. Lerner, B. Taskin, Slew merging region propagation for bounded slew and skew clock tree synthesis. *Trans. Very Large Scale Integr. (VLSI) Syst.* **27**(1), 1–10 (2019). <https://doi.org/10.1109/TVLSI.2018.2874572>

32. J. Li, L. Xiao, L. Li, H. Li, H. Liu, C. Wang, A low-cost error-tolerant flip-flop against SET and SEU for dependable designs. *Trans. Circ. Syst. I* **69**(7), 2721–2729 (2022). <https://doi.org/10.1109/TCI.2022.3168082>

33. Linear Technologies, Package Thermal Resistance Table, [https://www.cloudynights.com/ubbthreads/attachments/6565034-\\_Linear\\_Technology\\_Thermal\\_Resistance\\_Table.pdf](https://www.cloudynights.com/ubbthreads/attachments/6565034-_Linear_Technology_Thermal_Resistance_Table.pdf)

34. V. Melikyan, M. Martirosyan, A. Melikyan, G. Piliposyan, 14 nm educational design kit: capabilities deployment and future, in *Small Systems Simulation Symposium*, (2018)

35. A.K. Mishra, D. Vaithianathan, U. Chopra, Design and analysis of ultra-low power 18T adaptive data track flip-flop for high-speed application. *Int. J. Circuit Theory Appl.* **49**(11), 3733–3747 (2021). <https://doi.org/10.1002/cta.3124>

36. K.J. Nowka, G.D. Carpenter, E.W. MacDonald, H.C. Ngo, B.C. Brock, K.I. Ishii, T.Y. Nguyen, J.L. Burns, A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling. *J. Solid-State Circuits* **37**(11), 1441–1447 (2002). <https://doi.org/10.1109/JSSC.2002.803941>

37. J.M. Rabaey, *Low Power Design Essentials* (2009). <https://doi.org/10.1007/978-0-387-71713-5>

38. J.M. Rabaey, Digital integrated circuits: a design perspective, Chapter 7, in *Designing Sequential Logic Circuits* (2002), pp. 296–339

39. J.M. Rabaey, Digital integrated circuits: a design perspective, Chapter 10, in *Timing Issues in Digital Circuits* (2002), pp. 449–506

40. F.U. Rahman, V. Sathe, Quasi-resonant clocking: continuous voltage-frequency scalable resonant clocking system for dynamic voltage-frequency scaling systems. *J. Solid-State Circuits* **53**(3), 924–935 (2018). <https://doi.org/10.1109/JSSC.2017.2780219>

41. N. Sabu, K. Batri, Review of low power design techniques for flip-flops. *J. Pure Appl. Math.* **120**(6), 1729–1749 (2018)

42. V. Sathe, Quasi-resonant clocking: a run-time control approach for true voltage-frequency-scalability, in *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, (2014), pp. 87–92 <https://doi.org/10.1145/2627369.2627627>

43. G. Shin, E. Lee, J. Lee, Y. Lee, Y. Lee, A static contention-free differential flip-flop in 28nm for low-voltage, low-power applications, in *2020 IEEE Custom Integrated Circuits Conference (CICC)*, (2020), pp. 1–4 <https://doi.org/10.1109/CICC48029.2020.9075922>

44. B. Song, S. Choi, S.H. Kang, S.O. Jung, Offset-cancellation sensing-circuit-based nonvolatile flip-flop operating in near-threshold voltage region. *Trans. Circuits Syst. I* **66**(8), 2963–2972 (2019). <https://doi.org/10.1109/TCI.2019.2913009>

45. F. Stas, D. Bol, A 0.4-V 0.66-fJ/cycle retentive true-single-phase-clock 18T flip-flop in 28-nm fully-depleted SOI CMOS. *Trans. Circuits Syst. I.* **65**(3), 935–945 (2018). <https://doi.org/10.1109/TCSI.2017.2763423>
46. C.N. Sze, *ISPD 2010 High performance Clock Network Synthesis Contest* (2010), p. 143 <https://doi.org/10.1145/1735023.1735058>
47. V. Tirumalashetty, H. Mahmoodi, Clock gating and negative edge triggering for energy recovery clock, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, (2007), pp. 1141–1144. <https://doi.org/10.1109/ISCAS.2007.378251>
48. L. Touil, A. Hamdi, I. Gassoumi, A. Mtibaa, P. Agathoklis, Design of low-power structural fir filter using data-driven clock gating and multibit flip-flops. *J. Electr. Comput. Eng.* (2020). <https://doi.org/10.1155/2020/8108591>
49. M.Y. Tsai, P.Y. Kuo, J.F. Lin, M.H. Sheu, An ultra-low-power true single-phase clocking flip-flop with improved hold time variation using logic structure reduction scheme, in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (2018), pp. 1–4 <https://doi.org/10.1109/ISCAS.2018.8350985>
50. D. Vaithianathan, A.K. Mishra, T. Bhardwaj, V.J. Verma, B. Kaur, Power consumption and delay comparison of a modified TCFF with existing FF implemented using FinFET and load test circuit analysis, in *2021 IEEE Madras Section Conference (MASCON)*, (2021), pp. 1–5 <https://doi.org/10.1109/MASCON51689.2021.9563560>
51. H. You, J. Yuan, Z. Yu, S. Qiao, Low-power retentive true single-phase-clocked flip-flop with redundant-precharge-free operation. *Trans. Very Large Scale Integr. (VLSI) Syst.* **29**(5), 1022–1032 (2021). <https://doi.org/10.1109/TVLSI.2021.3061921>
52. Q. Yu, J. Gao, J. Wei, J. Li, K.C. Tan, T. Huang, Improving multispike learning with plastic synaptic delays. *Trans. Neural Netw. Learn. Syst.* (2022). <https://doi.org/10.1109/TNNLS.2022.3165527>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.