

Synthesis of Clock Networks with a Mode-Reconfigurable Topology

NECATI UYSAL and RICKARD EWETZ, University of Central Florida, Orlando, FL

Modern digital circuits are often required to operate in multiple modes to cater to variable frequency and power requirements. Consequently, the clock networks for such circuits must be synthesized, meeting different timing constraints in different operational modes. The overall power consumption and robustness to variations of a clock network are determined by the topology. However, state-of-the-art clock networks use the same topology in every mode, despite that timing constraints in low- and high-performance modes can be very different. In this article, we propose a clock network with a **mode-reconfigurable topology (MRT)** for circuits with positive-edge-triggered sequential elements. In high-performance modes, the MRT structure is reconfigured into a near-tree to provide the required robustness to variations. In low-performance modes, the MRT structure is reconfigured into a tree to save power. Non-tree (or near-tree) structures provide robustness to variations by appropriately constructing multiple alternative paths from the clock source to the clock sinks, which neutralizes the negative impact of variations. In MRT structures, OR-gates are used to join multiple alternative paths into a single path. Hence, the MRT structures consume no short-circuit power because there is only one gate driving each net. Moreover, it is straightforward to reconfigure an MRT structure into a tree topology using a single clock gate. In high-performance modes, the experimental results demonstrate that MRT structures have 25% lower power consumption than state-of-the-art near-tree structures. In low-performance modes, the power consumption of the MRT structure is similar to the power consumption of a clock tree.

CCS Concepts: • **Hardware** → **Clock-network synthesis**;

Additional Key Words and Phrases: Clock network synthesis, robustness to variations, near-tree topology, non-tree topology

ACM Reference format:

Necati Uysal and Rickard Ewet. 2022. Synthesis of Clock Networks with a Mode-Reconfigurable Topology. *ACM Trans. Des. Autom. Electron. Syst.* 27, 4, Article 38 (March 2022), 22 pages.
<https://doi.org/10.1145/3503538>

1 INTRODUCTION

Modern VLSI circuits are required to operate in low- and high-performance modes to deliver configurable frequency and power consumption. Synchronous VLSI circuits are synchronized by a clock network that delivers a clock signal from a clock source to the sequential elements or clock

This manuscript is based on the conference paper that was published at ISPD 2020 [26].

This work is supported by the National Science Foundation, under grant CCF-1755825.

Authors' address: N. Uysal and R. Ewet, University of Central Florida, 4000 Central Florida Blvd., Orlando, FL, 32816; emails: necati@knights.ucf.edu; rickard.ewetz@ucf.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

1084-4309/2022/03-ART38 \$15.00

<https://doi.org/10.1145/3503538>

sinks. *Clock skew* is the difference in the arrival time between a pair of clock sinks. There is a timing constraint in the form of a skew constraint between each pair of clock sinks that are only separated by combinational logic in the data and control paths. In each mode of operation, the skew constraints must be satisfied under variations to guarantee the functional correctness.

Constructing clock networks with small nominal skews is not too difficult [2, 4, 24, 25]. However, it is very challenging to meet tight skew constraints while a clock network is under the influence of **process, voltage, and temperature (PVT)** variations [3, 22, 23]. This is particularly challenging for circuits that are required to operate in low- and high-performance modes. Many recent studies are focused on constructing clock trees that utilize a combination of guardbands and useful skew to satisfy timing constraints under the variations in multiple modes [11]. However, for circuits with strict requirements on the clock frequency (in the high-performance modes), it may be impossible to satisfy the timing constraints using a clock network with a tree topology.

The overall power consumption and robustness to variations of a clock network are determined by the topology, which can be in the form of a tree, near-tree, or non-tree. Clock trees consume the least power but are vulnerable to variations. Clock networks with a non-tree or near-tree topology have multiple paths from the clock source to the clock sinks. When inserted appropriately, the alternative paths neutralize the negative impact of variations. However, clock networks with a non-tree topology (as clock meshes) may consume 3X–5X more power than a clock tree [20, 27, 30]. In the past decade, clock networks with near-tree topologies have been investigated, which promise significant improvements in robustness while the power consumption is similar to that of a clock tree [9, 12, 14, 16, 17, 28]. In [14, 16], and [28], near-tree structures were constructed by inserting cross-links. In [12], multiple tree structures were fused together in order to create a multilevel fusion tree. In [9], a locally merged structure was formed by fusing subtrees at internal nodes of a clock tree. The fusion was performed at internal nodes instead of at the clock sinks to reduce hardware overheads. A drawback of all near-tree structures is that there are multiple gates driving the same net of wires, which may result in short circuit current. Moreover, the same topology is used in every operational mode, despite that the robustness provided by a near-tree is only required in the high-performance modes. It is easy to understand that a clock network with a tree topology could easily meet the timing constraints with much smaller power in the low-performance modes.

In this article, a clock network with a **mode-reconfigurable topology (MRT)** is proposed for circuits that only have positive-edge-triggered sequential elements and are required to operate in the low-performance mode and high-performance mode. The main contributions of this article are summarized, as follows:

- *Reconfigurable Topology.* The proposed MRT structure can reconfigure the topology of the clock network based on the operational mode. In the high-performance mode, the MRT structure delivers the clock signal using a near-tree topology to provide robustness to variations. In the low-performance mode, power is saved by reconfiguring the clock network to deliver the clock signal using a tree topology.
- *Circuit Innovations.* The reconfiguration of the topology of an MRT structure is facilitated by OR-gates and a single clock gate. The OR-gates are used to construct alternative paths to each clock sink, which improves the robustness to variations. The advantage of using OR-gates to construct redundant paths is that no short circuit current is introduced, as there is only a single gate driving each net of wires.
- *Automated Synthesis Methodology.* We propose an automated synthesis methodology to construct MRT structures. The methodology balances hardware overhead and robustness to variations.

- *Experimental Validation.* The experimental results show that MRT structures provide higher robustness to variations and up to 25% lower power consumption than state-of-the-art near-tree structures in the high-performance mode. In the low-performance mode, the power consumption is 35% lower due to the ability of reconfiguring the topology into a tree.

The remainder of the article is organized as follows: Preliminaries are detailed in Section 2. Problem formulation is given in Section 3. Previous studies are reviewed in Section 4. The proposed MRT structure is given in Section 5 and the methodology is given in Section 6. The experimental results are presented in Section 7. The article is concluded in Section 8.

2 PRELIMINARIES

Timing Constraints. Synchronous circuits are synchronized by delivering a clock signal from a clock source to flip-flops (or clock sinks). There is a setup and hold time constraint between any pair of **flip-flops (FFs)** that are only separated by combinational logic in the data and control paths. The clock signal must be delivered, meeting setup and hold time constraints for the functional correctness of the circuit. The setup and hold time constraints between the launching flip-flop FF_i and the capturing flip-flop FF_j are formulated, as follows:

$$t_i + t_i^{CQ} + t_{ij}^{max} + t_j^S + \delta_i^{ij} \leq t_j + T - \delta_j^{ij}, \quad (1)$$

$$t_i + t_i^{CQ} + t_{ij}^{min} - \delta_i^{ij} \geq t_j + t_j^H + \delta_j^{ij}, \quad (2)$$

where t_i and t_j are the arrival times of the clock signal to the FF_i and FF_j , respectively. t_i^{CQ} is the clock to output delay of the launching flip-flop; T is the clock period; t_j^S and t_j^H are the setup time and hold time of FF_j , respectively. t_{ij}^{max} (t_{ij}^{min}) is the maximum (minimum) delay through the combinational logic between FF_i and FF_j . δ_i^{ij} and δ_j^{ij} are the timing deteriorations introduced between FF_i and FF_j by variations. The magnitude of the timing deteriorations is correlated with the distance between the clock sinks in the topology of the clock network [18]. For circuits with multiple operational modes, the timing constraints in Equations (1) and (2) are required to be satisfied in all modes. The terms in Equations (1) and (2) have different values at each operational mode due to different clock frequencies and supply voltages used.

The clock skew between the pair of clock sinks is defined to be equal to $skew_{ij} = t_i - t_j$. Let δ^{ij} be the total magnitude of the timing deteriorations that are introduced between FF_i and FF_j , i.e., $\delta^{ij} = \delta_i^{ij} + \delta_j^{ij}$. Using $ub_{ij} = T - t_i^{CQ} - t_{ij}^{max} - t_j^S$ and $lb_{ij} = t_j^H - t_i^{CQ} - t_{ij}^{min}$, the setup and hold time constraints in Equations (1) and (2) can be reformulated as explicit skew constraints, as follows:

$$lb_{ij} + \delta^{ij} \leq skew_{ij} \leq ub_{ij} - \delta^{ij} \quad (3)$$

Let B be a lower bound on $|lb_{ij}|$ and $|ub_{ij}|$. Note that lb_{ij} is negative and ub_{ij} is positive for the circuits that are focused on this paper. Similarly, let D be an upper bound of the timing deteriorations δ^{ij} for a clock network with a tree topology. D is determined after the clock network is constructed and the timing deteriorations δ^{ij} are known. This can be achieved by performing timing analysis for each operational mode while applying various forms of variations and recording the maximum δ^{ij} . Consequently, the skew constraints in Equation (3) can be reformulated, as follows:

$$D - B \leq skew_{ij} \leq B - D. \quad (4)$$

Meeting the explicit skew constraints under the nominal conditions is easy ($D = 0$). When $D < B$, it is typically possible to construct a clock tree that satisfies the timing constraints under variations.

When $D > B$, it is in general necessary to construct a clock network with a near-tree topology.¹ Near-tree clock networks have a tighter upper bound on δ^{ij} . The bound is tighter because all pairs of clock sinks with timing constraints are located closer in the near-tree topology [12]. The near-tree structures are commonly synthesized with zero skew to maximize the robustness to variations [22, 23].

Dynamic Voltage and Frequency Scaling (DVFS). Circuits with multiple modes utilize a lower clock frequency in the low performance modes. The scaled down clock frequency allows the supply voltage to be scaled down to save power without introducing any timing violations. The dynamic power consumption P is calculated, as follows [29]:

$$P = \alpha_{sw} \cdot C_L \cdot V_{DD}^2 \cdot f, \quad (5)$$

where α_{sw} is the activity factor, C_L is the switching capacitance, V_{DD} is the supply voltage, and f is the clock frequency.

When the clock frequency is reduced in the low-performance modes, timing margins are introduced in the limiting setup and hold time constraints. These timing margins allow the supply voltage to be scaled down until the maximum propagation delay through the combinational logic becomes too long. The relationship between the supply voltage and the propagation delay can be formulated as follows [21]:

$$t_{pd} \propto \frac{V_{dd}}{(V_{dd} - V_t)^2}, \quad (6)$$

where t_{pd} is the propagation delay of the combinational logic, V_t is the threshold voltage. It can also be understood from the Equation (6) that scaling up the supply voltage reduces the propagation delay, which introduces timing margins in the setup and hold time constraints. Consequently, the supply voltage can be scaled up to meet the timing constraints in Equations (1) and (2), which results in increasing the power consumption. Furthermore, several recent studies attempt to squeeze out additional power savings by performing clock tree optimization, where multiple modes are simultaneously optimized to improve both timing and power [7, 18].

3 PROBLEM FORMULATION

In this article, we solve the problem of constructing clock networks that connect a clock source to a set of clock sinks (with specified xy-locations) using wires, buffers, clock gates, and OR-gates from a given technology library. This article is focused on constructing clock networks for circuits that only have positive-edge triggered sequential elements and are required to operate either in a low-performance mode or in a high-performance mode.

The clock frequency and the timing constraints are different for each operational mode. Consequently, clock networks must meet the setup and hold time constraints in Equations (1) and (2) under PVT variations for each mode of operation. Moreover, the transition time constraints are required to be satisfied. The objective is to minimize the weighted clock power (P_T) using the cost metric as follows:

$$P_T = \beta P_H + \gamma P_L, \quad (7)$$

where P_H and P_L are the power consumption in high-performance mode and low-performance mode, respectively. β and γ are user defined parameters, which are used to balance P_H and P_L in Equation (7).

¹Useful skew can be leveraged to enable clock trees to be constructed when D is only slightly larger than B [10].

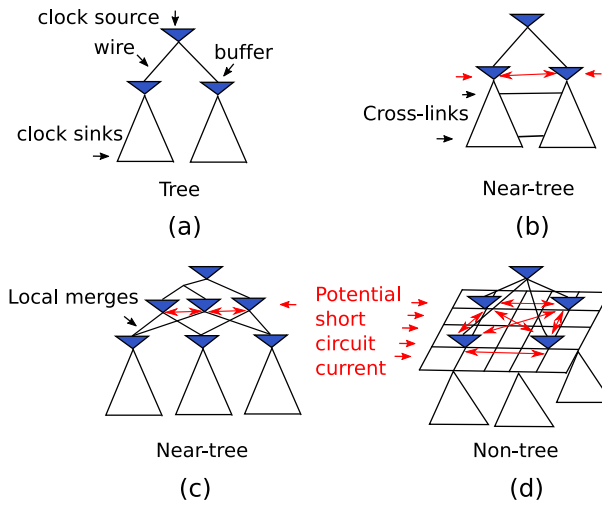


Fig. 1. Clock networks with different topologies.

4 PREVIOUS STUDIES

The topology of a clock network can be in the form of a tree [2, 4, 5, 24, 25], a near-tree [9, 14, 16], or a non-tree [20, 27, 30], as illustrated in Figure 1. Near-tree structures can be in the form of clock trees with cross-links, clock spines [19], multilevel fusion trees [12], and locally merged structures [9]. In Table 1, we compare the properties of clock networks with different topologies.

The tradeoff between robustness and power consumption for tree and non-tree (or near-tree) topologies is well known [9, 12, 14, 16, 20, 30]. The non-tree (or near-tree) structures have multiple alternative paths from the clock source to each clock sink, which neutralizes the negative impact of variations. Naturally, the multiple paths introduce hardware overheads with respect to a tree topology. The current trend is to insert the redundancy at internal nodes of a tree structure to balance the robustness improvements with the hardware overheads [9, 12]. Clearly, near-tree structures seem advantageous to non-tree structures as clock meshes. Nevertheless, near-tree structures are not widely (or at all) adopted in the industry. This is due to that (i) near-tree structures introduce short circuit current and (ii) near-tree structures cannot be constructed using the tree construction algorithms that are integrated into the EDA tools. These challenges stem from that near-tree structures have multiple gates driving the same net of wires, as shown in Figure 1. One of the reasons why clock meshes have been successfully adopted in the industry is that symmetric clock trees are traditionally used to drive the mesh grid, which is easy to construct. Moreover, there have already been substantial investments in specialized EDA tools for the synthesis and analysis of such mesh structures. Nevertheless, the concept of sink-splitting was recently proposed to allow near-tree structures to be constructed using tree construction algorithms [12]. The technique involves splitting sinks (or the input pins to buffers) into multiple virtual sinks (or pins). Next, after a tree structure has been constructed to deliver the clock signal to each of the virtual sinks, the virtual sinks are fused to form a near-tree structure. An intuitive idea to reduce the power consumption in the low-performance modes is to turn-off part of the non-tree (or near-tree) structure using clock gates while still guaranteeing that the clock signal is delivered using at least one path from the clock source to each clock sink. However, this is impossible to perform using state-of-the-art non-tree (or near-tree) structures because it would create static short circuit current.

Table 1. Comparison between Clock Networks with Tree, Near-Tree, and Non-Tree Topologies

Topology	Work	Robustness to variations	Power	Based on tree structure	Compatible with	
					Voltage scaling	Reconfiguration
Tree	[2, 4, 25]	low	small	Yes	Yes	No
Near-tree	[9, 12, 16]	high	medium	No	Yes	No
Non-tree	[20, 27, 30]	high	very large	No	Yes	No
MRT (near-tree)	This work	high	medium	Yes	Yes	Yes

In this article, we propose a clock network with a **mode reconfigurable topology (MRT)** with properties as labeled in Table 1. The MRT structure has similar performance as near-tree structures in earlier studies but multiple paths are joined using OR-gates instead of wires. Consequently, there is only one gate driving each net of wires. Therefore, it is easy to understand that the MRT structure can be reconfigured into a tree topology by gating the clock signal in part of the structure. Moreover, the MRT structures can be constructed using the tree construction algorithms that are adopted in the EDA tools. Consequently, the MRT structure is based on tree structures.

5 PROPOSED MRT STRUCTURE

In this section, we introduce the properties of the proposed MRT structure. The overview of the MRT structure is given in Section 5.1. In Section 5.2, it is explained how MRT structure in the form of near-tree topology improves robustness to variations. In Section 5.3, we explained how MRT structures save power in the low-performance mode.

5.1 Overview of the MRT Structure

In this section, the proposed MRT structure is explained using the Figure 2. The MRT structure is constructed to operate in two modes using buffers, wires, OR-gates and a clock gate. The structure consists of two top-level trees that are joined using OR-gates to drive the bottom-most subtrees. The input to the MRT structure is a mode control signal and the clock signal from the clock source. The mode control signal specifies the mode that the circuit should operate. In the high-performance mode 1, the clock gate is transparent and the entire MRT structure is used to deliver the clock signal to the clock sinks, i.e., the topology is in the form of a near-tree. In the low-performance mode 2, the clock signal is gated and the part of the MRT structure that is used to deliver the clock signal to the clock sinks is in the form of a tree topology. In Figure 2, a few gates and subtrees are labeled ‘1’ and ‘1 2’ to indicate the modes where the respective components are used to deliver the clock signal.

5.2 Improving the Robustness in High-Performance Modes

In the high-performance mode, the robustness to variations is improved by the two top-level trees that are joined together using the OR-gates. However, OR-gates only improve the robustness of the positive-edge of the clock signal. Nevertheless, positive-edge triggered sequential elements have loose timing constraints on the negative-edge of the clock signal. Therefore, it can be concluded that the MRT structure improves the overall robustness to variations.

The robustness of the positive-edge is improved because the OR-gates are implemented by NAND-gates with one INV-gate connected to each their inputs, as illustrated in the bottom right part of Figure 2. The two-input NAND-gate has two parallel PMOS transistors that may charge the net attached to the output of the NAND-gate. Under nominal conditions, the MRT structure is designed such that the clock signal arrives at the same time to the separate input pins of each

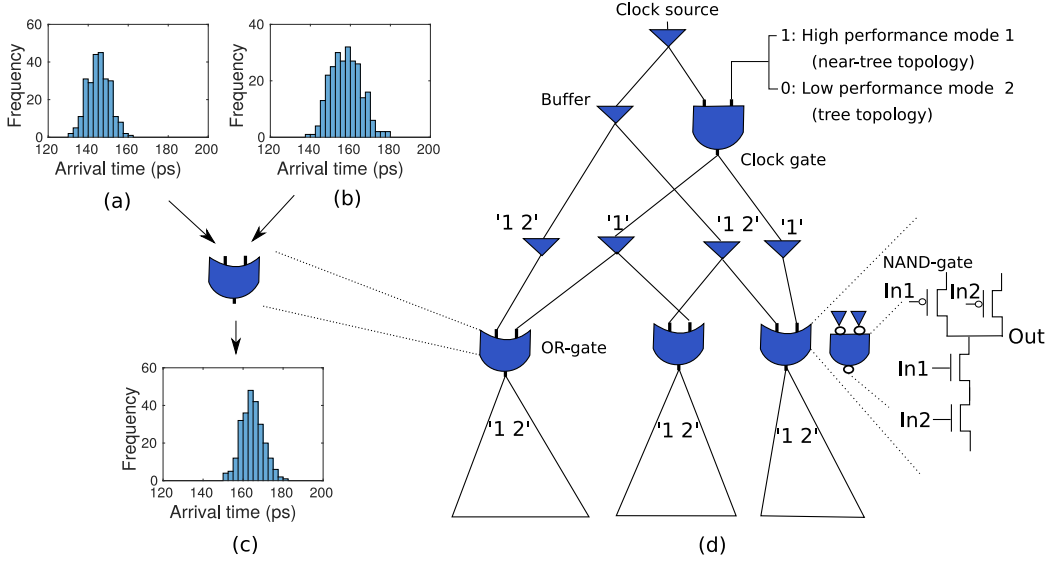


Fig. 2. The arrival time distributions (for the positive-edge of the clock signal) at different points with respect to an OR-gate are shown in (a), (b), and (c). The variance of the arrival time distribution at the output of the OR-gate is tighter than that of the variance at the inputs, which demonstrates that OR-gates improve the robustness to variations. (d) The proposed MRT structure is constructed using OR-gates to provide robustness to variations.

OR-gate. Hence, both PMOS transistors in the NAND-gate will equally contribute to charging the downstream net. Under variations, it is expected that the arrival time of the clock signal to the separate input pins will be different. The end-result is that the PMOS transistor that is turned-on first (last) will charge the net more (less) compared within the nominal case. A simple but effective model to estimate the robustness improvement of an OR-gate is to approximate the effective arrival time of the clock signal to the OR-gate with the mean of the two separate arrival times, i.e., an OR-gate will improve the robustness to variations by averaging out the negative effect of the variations. Of course, the model only holds when the last PMOS transistor to be turned-on is activated before the net downstream of the OR-gate is completely charged by the other PMOS transistor. Therefore, the proposed OR-gate is preferred to the traditional OR-gate consisting of a NOR-gate connected in series with an INV-gate. The explanation is that the capacitance of the internal node in the traditional OR-gate is smaller than the capacitance of the downstream subtree, which means that the internal node will be charged faster, and fewer variations can be neutralized.

To confirm that OR-gates improve the robustness of the delivery of the positive-edge of the clock signal, the structure in the figure is simulated using NGSPICE [15] while applying various forms of correlated variations such as wire width, channel length, supply voltage, and temperature variations using the Monte Carlo framework in [9]. More details about the variations are provided in Section 7. The arrival time distributions at the inputs of an OR-gate are shown in (a) and (b) of Figure 2. It is concluded that the OR-gates improve the robustness to variations because the variance of the arrival time distribution at the output of the OR-gate is close to half of the variance of the distribution above the OR-gate, which is shown in Figure 2(c).

In terms of the negative-edge of the clock signal, it can be understood that the OR-gates act as a max operator using the same simple model. Although we are targeting designs with only

Table 2. Comparison between DVFS and Topology Reconfiguration Combined with DVFS

Technique	V_{dd}	C_{clk}	Power
DVFS	very small	full	small
DVFS + Reconfiguration	small	reduced	very small

positive-edge triggered sequential elements in this article, we experimentally observed that the MRT structures also deliver the negative-edge of the clock signal with higher robustness than a clock tree (see Section 7.1.4). The explanation is that the max operator effectively slows down the clock signal on paths where the propagation delay has been sped-up by variations more than it further slows down the propagation delay of paths that have become slower due to variations. Nevertheless, we intend to perform further analyses of the impact on the setup and hold times for circuits with negative-edge and dual-edge triggered sequential elements in our future work.

5.3 Reducing Power in Low-Performance Modes

To save power in the low-performance mode, the MRT structure can be reconfigured from near-tree topology to a tree topology based on the active mode. The reconfiguration can be performed without introducing any short-circuit current because each net of wires is connected to only a single driving gate. In fact, this is the first near-tree (or non-tree) structure that can be configured based on the active mode. However, the reconfiguration introduces a tradeoff between supply voltage scaling and clock network switching capacitance, which is shown in Table 2.

The dynamic power consumption of a VLSI circuit is calculated, as follows:

$$P = C_{comb} \cdot V_{DD}^2 \cdot f \cdot \alpha_{comb} + C_{clk} \cdot V_{DD}^2 \cdot f \cdot \alpha_{clk}, \quad (8)$$

where f is the clock frequency; C_{comb} and C_{clk} are the total capacitance of the combinational logic and the clock network, respectively; α_{comb} and α_{clk} are the activity factors of the combinational logic and clock network, respectively; and V_{DD} is the supply voltage.

Traditional DVFS technique is performed by scaling down the supply voltage to the minimum possible value that the timing constraints are guaranteed to be satisfied. For MRT structures, we propose to combine topology reconfiguration with DVFS in the low-performance mode. First, the MRT structure is reconfigured into a tree topology. Second, DVFS is applied to the MRT structure.

The reconfiguration of the topology into a clock tree reduces the clock tree switching capacitance (C_{clk}) by turning off a large portion of the clock network. However, the reconfiguration may introduce small nominal skews between pairs of clock sinks because the MRT structure has OR-gates with a different number of inputs, which have different propagation delays. Therefore, the timing margins available for voltage scaling are reduced by the reconfiguration. Consequently, the supply voltage will be slightly higher compared with only applying voltage scaling. Nevertheless, the overall power is expected to be significantly lower. The explanation is that the propagation delay of any gate is proportional to Equation (6). Therefore, large timing margins are required to only slightly reduce the supply voltage when the supply voltage V_{DD} is close to the V_t . Consequently, the reduction in the C_{clk} is expected to translate into large power savings than reducing V_{DD} slightly more.

6 METHODOLOGY

The flow for constructing the proposed MRT structures is shown in Figure 3. The flow is illustrated with an example in Figure 4. MRT structures are constructed bottom-up stage-by-stage using a

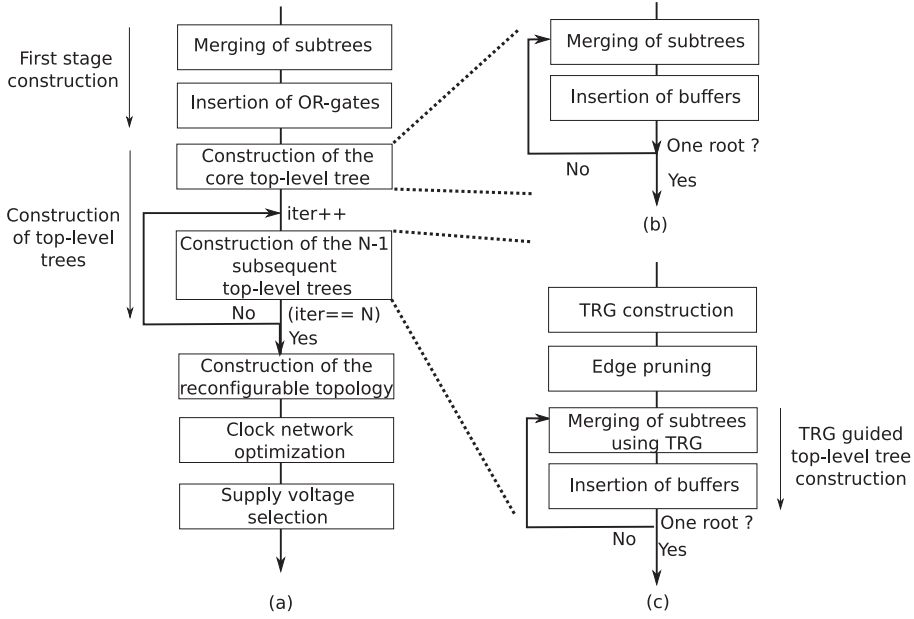


Fig. 3. (a) Flow for constructing MRT structures. (b) Flow for construction of the core top-level tree. (c) Flow for construction of the $N - 1$ subsequent top-level trees.

modification of the classic zero-skew **Deferred-Merge Embedding (DME)** paradigm [3, 5, 6]. A stage consists of a device (a buffer or an OR-gate) driving a subtree of wires connected to the stage-sinks. The stage-sinks of the first stage are the clock sinks. The stage-sinks of any other stage are the input pins of the devices driving the previous stage. The bottom-up construction phase is followed by a top-down embedding of internal nodes using the DME paradigm.

MRT structures consist of first-stage subtrees that are driven by N top-level trees, where $N \geq 2$ is a user-specified parameter. The first top-level tree is constructed with the objective of minimizing wirelength while connecting each clock sink to the clock source. The first top-level tree is referred to as the core top-level tree. The remaining $N-1$ top-level trees are constructed to improve the robustness (see details in Section 6.3). The N top-level trees are joined to drive the first-stage subtrees using OR-gates. We choose to insert OR-gates to drive the subtrees of the first stage because such procedure is known to balance robustness and capacitive cost [9]. The robustness provided by inserting redundancy is higher when placed closer to the clock sinks. At the same time, inserting redundancy closer to the clock sinks introduces larger hardware and performance overheads. The roots of the N top-level trees are connected together using a clock gate in order to enable the topology reconfiguration.

The flow in Figure 3 shows that the first stage of an MRT is constructed by merging subtrees (see Section 6.1.1) and inserting OR-gates to drive the subtrees (see Section 6.2). Next, the construction of the core top-level tree step is performed. This is followed by performing the construction of $N-1$ subsequent top-level trees step (see Section 6.3). The top-level trees are subsequently joined, gated, and routed to the clock source (see Section 6.4). After constructing the reconfigurable topology, the nominal skew of the MRT is tuned close to zero using a clock network optimization step (see Section 6.5). Last, DVFS is applied to determine the supply voltage for both modes of operation (see Section 6.6).

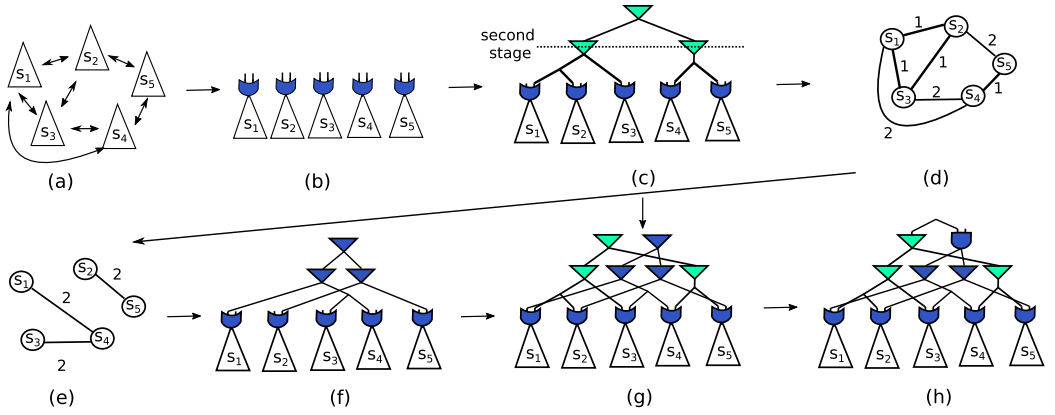


Fig. 4. Example flow for the construction of an MRT structure when $N = 2$. (a) First-stage subtrees. (b) Driver devices are inserted. (c) Core top-level clock tree is constructed. The driver devices used to construct the core-top level tree are colored in green. (d) The Topology Relation Graph (TRG) is constructed with respect to the topology. (e) The TRG after edge pruning is applied. (f) The second top-level clock tree is constructed using TRG. (g) The clock network after the construction of top-level trees shown in (c) and (f). (h) The clock network after constructing the reconfigurable topology.

6.1 Zero-Skew Tree Construction [2, 4, 24, 25]

In this section, it is explained how a **zero-skew tree (ZST)** with buffers can be constructed based on iteratively performing a *merging of subtrees phase* and an *insertion of buffers phase*, which is shown in Figure 3(b). The method is based on techniques proposed in [2, 5, 24], and [25].

6.1.1 Merging of Subtrees. The merging of subtrees step is used to construct a set of zero skew subtrees from a set of stage-sinks using the DME algorithm. The construction is based on iteratively merging pairs of smaller subtrees into a larger subtree using zero-skew merges. A zero-skew merge joins a pair of subtrees with the smallest possible wirelength while ensuring that the delay to all the sinks in the formed subtree are equal under the Elmore delay model. The merging process is facilitated by a **nearest neighbor graph (NNG)** [6]. An NNG is used to capture the subtrees (or stage sinks) and the wirelength distances between the pairs of subtrees. Iteratively, the pair of subtrees that require minimum amount of wirelength to be joined is selected to be merged. After a pair of subtrees have been joined, the transition time is evaluated. If the transition time constraint is violated, the subtree pair is unmerged and both subtrees are locked from further merging. Otherwise, the newly formed subtree is allowed to be joined with additional subtrees. The process is repeated until all subtrees are locked from further merging. Please refer to [2, 4, 5, 24], and [25] for further technical details.

6.1.2 Insertion of Buffers. The input to the buffer insertion is a set of locked subtrees from the merging of subtrees step. For each subtree, the minimum-sized buffer that can drive the locked subtree (without violating the transition-time constraint) is selected to be inserted at the root. Next, a piece of stem wire is inserted between the subtree and the buffer [5].

6.2 Insertion of OR-Gates

In this section, we explain the insertion of OR-gates step. The input to this step are the first-stage subtrees. For each subtree, the minimum sized OR-gate (with N input pins) that can drive the subtree without violating the transition time constraint is inserted at the root. The parameter N

is limited to 4 in order to limit the size of the NMOS transistors within each NAND-gate (inside each OR-gate). A set of first stage subtrees are shown in Figure 4(a). After the OR-gate insertion step, the resulting subtrees are shown in Figure 4(b).

6.3 Construction of Top-Level Clock Trees

In this section, we explained how the N top-level trees in an MRT structure are constructed. The core top-level tree is constructed using the traditional zero-skew clock tree construction algorithm, which involves iteratively merging subtrees with the objective of minimizing wirelength. Next, the remaining $N-1$ top-level trees are constructed with the goal of improving the robustness to variations. This is achieved by constructing the subsequent $N-1$ top-level trees with the following two objectives:

- (i) Clock sinks (with skew constraints between them) that are distant in the base (first) top-level tree should be placed close in at least one of the subsequent top-level trees.
- (ii) Subtrees that are close (topology-wise) in the core top-level tree should not be close in any subsequent top-level tree.

The motivation for the first objective is that larger variations are introduced in skew constraints between clock sinks that are distant in the topology. Therefore, the robustness can be significantly improved by placing such clock sinks (or subtrees) close in a subsequent top-level tree. The motivation for the second objective is that clock sinks that are close in the initial tree topology have no need of improving the robustness to variations. Therefore, the construction of such top-level trees should be avoided in order to minimize hardware overheads. Moreover, the construction of clock networks that are larger than necessary may actually degrade the overall robustness, as larger clock networks are more vulnerable to variations [9]. We propose to achieve the two aforementioned objectives by introducing a **topology relation graph (TRG)**. Next, the TRG is used to guide the construction of the last $N-1$ top-level trees. The TRG is used to capture both skew constraints and the distance between pairs of subtrees in the tree topology. The first objective is achieved by encouraging subtree pairs with large-edge weights in the TRG to be joined. The second objective is achieved by pruning edges in the TRG.

The details of the TRG construction are explained in Section 6.3.1. The edge pruning step is given in Section 6.3.2. The TRG-guided clock tree construction is provided in Section 6.3.3. These steps are performed iteratively to construct the last $N-1$ top-level trees.

6.3.1 Construction of Topology Relation Graph. A TRG $G = (V, E)$ is constructed with respect to the first-stage subtrees. The vertices V represent the first-stage subtrees. The weighted edges E represent the distance in the topology between pairs of subtrees that are sequentially related. A pair of clock sinks are sequentially related if there is a skew constraint between them. A pair of subtrees are called *sequentially related* if they correspondingly contain a pair of sequentially related clock sinks. The edge weight between sequentially related subtrees is defined to be equal to the minimum number of buffers that must be traversed to find the **closest common ancestor (CCA)** of the subtree pair. For example, in Figure 4(d), the edge weight between s_2 and s_3 is one because only one CCA buffer is required to be traversed. On the other hand, the edge weight between s_2 and s_5 is 2 because two buffers are required to be traversed. The arrows in the Figure 4(a) illustrate that the corresponding first-stage subtree pairs are sequentially related. The TRG with respect to the topology of the core top-level tree in Figure 4(c) is illustrated in Figure 4(d).

After multiple top-level trees have been constructed, it can be understood that each subtree pair may have multiple CCAs. When forming the TRG for the construction of the third or fourth top-level tree, a candidate edge-weight value is computed with respect to each of the previous

top-level trees. Next, the weight for each edge in the TRG is set to the minimum of the candidate values.

6.3.2 Edge Pruning. In this section, the edge pruning technique is explained. The edge pruning is performed by removing all edges in a TRG with a weight less than or equal to one. This corresponds to avoiding joining subtrees that are joined in the second stage of the initial tree topology. Figure 4(e) shows the resulting TRG after performing edge pruning in the TRG of Figure 4(d). After pruning the edges, there may exist vertices without any incident edges, which means that the clock signal is already delivered with sufficient robustness. Consequently, there is no need for the corresponding subtree to participate in the subsequent top-level tree construction. This is facilitated by reducing the number of input pins of the corresponding OR-gate by one.

6.3.3 TRG Guided Tree Construction. In this section, we explained how the TRG is used to guide the construction of the top-level trees. The input to the TRG-guided clock tree construction step is one input pin from each OR-gate that has been inserted to drive the first-stage subtrees. The output is a top-level tree driving the input pins. The top-level tree is constructed by iteratively performing the merging of subtrees and the insertion of buffers steps, which is shown in Figure 3(c). However, when selecting subtrees to be merged in the NNG, both the required wirelength and topology distance are considered (see details below). Moreover, if no edge exists between a pair of vertices in the TRG, the corresponding subtrees in the NNG cannot be merged. The top-level tree constructed by the TRG guided tree construction is shown in Figure 4(f). The clock network structure that is obtained after the construction of two top-level trees are shown in Figure 4(g).

The conventional zero-skew tree construction methodology is based on joining the subtrees that require the minimum wirelength to be merged. In the TRG-guided top-level tree construction, the merging of the subtrees is based on a cost function that accounts for both wirelength and distance in the topology. Let the cost of merging a subtrees i and j be denoted c_{ij} and defined, as follows:

$$c_{ij} = \alpha d_{ij} - (1 - \alpha) p_{ij} \quad (9)$$

where d_{ij} is the wirelength required to join the subtrees i and j using a zero-skew merge. p_{ij} is the corresponding edge weight in the TRG. α is a user defined parameter to balance the importance of the two objectives. It can be understood that setting α to 1 would result in constructing a top-level tree using the traditional zero-skew metric. In our framework, we set α to be small (0.01) such that the cost metric always prioritizes topological distance over the wirelength. The wirelength component is used as a tie-breaker when there exist multiple edges with the same weight (p_{ij}).

Note that when two subtrees are merged, the corresponding vertices in the TRG must be merged into a single vertex. The incident edges of the two vertices are inserted between the newly formed vertex and the corresponding adjacent vertices (without modifying the edge weights). This may result in two parallel edges with different edge weights that are introduced between the newly formed vertex and a single adjacent vertex. For such cases, the parallel edges are replaced with a single edge with a weight equal to the minimum of the two edge weights.

6.4 Construction of the Reconfigurable Topology

In this section, we explained how the N constructed top-level trees are joined into an MRT structure. The top-level trees must be joined such that the MRT structure can be reconfigured between a tree and a near-tree topology. The reconfiguration is performed by gating the clock signal in part of the structure using a clock gate. In contrast with traditional clock gating that aims to completely turn off a portion of the clock network, the reconfiguration based clock gating is performed such that there still exists at least one path from the clock source to each clock sink. Let the constructed top-level trees be denoted t_1 to t_N . To facilitate the reconfiguration between the tree and

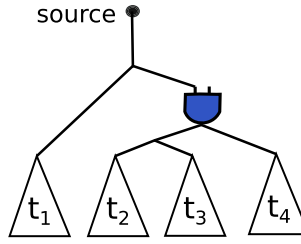


Fig. 5. The reconfigurable topology.

near-tree topology, we proposed to connect the top-level trees as illustrated in Figure 5. In the high-performance mode, the MRT structure has a near-tree topology and all the top-level trees t_1 to t_N deliver the clock signal to clock sinks. When the clock signal is gated in the low-performance mode, the top-level trees t_2 to t_N are turned off and only the core top-level tree t_1 delivers the clock signal to all clock sinks. Note that t_1 is the only top-level tree that is guaranteed to have a path to all the first-stage subtrees.

The structure in the figure is constructed by joining t_2 to t_N using zero-skew merges. Next, we insert a clock gate to drive the resulting tree before merging the clock gate with the core top-level tree t_1 using a zero skew merge. Finally, we connect the resulting structure to the clock source. Of course, the buffers are inserted to balance delay if a zero-skew merge cannot be performed without violating the transition time constraints. This results in the MRT structure in Figure 4(h) which is formed from the two top-level trees in Figure 4(g). Next, a clock network optimization step in Section 6.5 is performed to balance the skew in the structure.

6.5 Clock Network Optimization

Clock network optimization (CNO) is performed to minimize the nominal skew in a clock network. It is typically very difficult to construct large clock networks with small nominal skew without utilizing CNO. For MRT structures, clock network optimization is performed with the objectives of: (i) minimizing the nominal skew close to zero and (ii) minimizing the skew between input pins of the same OR-gate, which improves the robustness to variations.

State-of-the-art clock tree optimization techniques are based on specifying and realizing delay adjustments to remove the timing violations, which is also essential to meet the useful skew constraints [13]. The specified delay adjustments are realized by inserting delay buffers and detour wires. For MRT structures, clock network optimization is performed using a two-phase approach. In the first phase, the input pins of the OR-gates are viewed as the clock sinks and the skew between pins of the same OR-gate is minimized by specifying and realizing delay adjustments, which improves the robustness of the MRT structures to variations. In the second phase, the nominal skew is minimized by specifying and realizing delay adjustment below the OR-gates. Note that it is easy to adapt clock-tree optimization algorithms to the MRT structure because there is only a single gate driving each net, which enables conventional EDA algorithms and tools to be used.

6.6 Supply Voltage Selection

In this section, it is explained how the supply voltage is determined at each operational mode. The objective is to select the supply voltage that meets the timing constraints with the minimum possible power consumption while operating with the specified clock frequency for the operational mode.

After applying CNO to the MRT structure, 250 Monte Carlo simulations are performed to obtain the upper bound on the timing deteriorations (D) that the MRT structure achieves. Next, we tighten

Table 3. Benchmarks in [8]

Circuit (name)	Sinks (num)	Skew constraints (num)	Clock period (ps)		Clock cycle for guardbands (%)
			T_H	T_L	
s1423	74	78	200	1,000	10
usbf	1,765	33,438	200	1,000	10
dma	2,092	132,834	200	1,000	10
pci bridge32	3,582	141,074	200	1,000	10
ecg	7,674	63,440	200	1,000	10
des peft	8,808	17,152	200	1,000	10
aes	13,216	53,382	200	1,000	10

Table 4. The Properties of the Structures

Structure	Redundancy insertion using	Synthesis is guided by	Topology
Tree [2, 5, 25]	'-'	NNG	Tree
Near-Tree-SS [9]	Sink splitting	NNG + Local merges	Near-tree
Near-Tree-OR	OR-gates	NNG + Local merges	Near-tree
MRT	OR-gates	NNG + TRG	Near-tree

the timing margins by the amount of D to provide guardbands to variations. Finally, we perform DVFS by applying a binary search to determine the minimum supply voltage that meets the timing constraints. In our framework, the supply voltage is allowed to be specified in the range of V_t to $1.02V$. Note that the supply voltage can never be scaled down too close to the threshold voltage because there are no excessive timing margins available in the evaluated designs.

7 EXPERIMENTAL RESULTS

The proposed methodology was implemented in C++ and experimental evaluation was performed on a quad-core 3.4-GHz Linux machine with 32 GB of memory. The properties of the buffers, the OR-gates, and the wires were obtained from a 45-nm technology library [22]. The benchmarks in [8] were used to synthesize our clock networks. The details about benchmark circuits (number of clock sinks, number of skew constraints, and clock period) are shown in Table 3. The transition time constraint for all the structures was set to 100 ps. The clock periods correspond to a clock frequency of 5GHz and 1GHz in the high- and low-performance modes, respectively. In each mode of operation, a portion of the clock cycle was reserved for guardbands to variations, which is shown in the “Clock cycle for guardbands” column in Table 3.

To demonstrate the effectiveness of the proposed framework, four different structures are constructed and evaluated. The properties of each structure are summarized in Table 4. For each structure, the techniques used to insert redundancy and guide the synthesis process are reported along with the topology. Specifically, we construct one tree structure (labeled “Tree”), two near-tree structures (labeled “Near-tree-SS” and “Near-Tree-OR”), and the proposed MRT structure (labeled “MRT”). The tree structures were constructed using the methodology in Section 6.1, which is similar to that for the zero-skew clock trees in [2, 5], and [25]. The Near-Tree-SS structures were constructed to mimic the locally-merged structures in [9], which are illustrated in Figure 1(c). The locally merged structures were selected because they were reported to outperform clock trees with cross-links and clock meshes in [9]. The Near-Tree-OR structures were constructed to mimic the Near-Tree-SS structures. However, alternative redundant paths were joined using OR-gates instead of using sink-splitting. The MRT structures were constructed using the proposed flow in Figure 3(a).

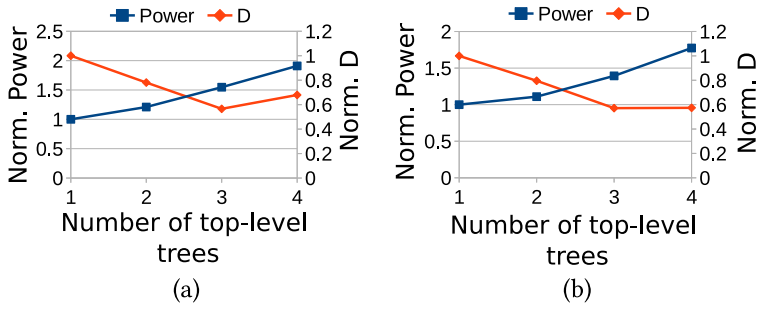


Fig. 6. Evaluation of MRT structures with different number of top-level trees on the circuits (a) *dma* and (b) *aes*. The performance is evaluated in terms of power and robustness to variations. The clock network structure with 1 top-level tree is equivalent to the traditional clock tree.

Based on the experiments performed in Section 7.1, the MRT structures were constructed using two top-level trees, i.e., the MRT structures were obtained using $N = 2$.

The structures are evaluated in terms of power consumption, supply voltage, and timing quality that are obtained in high- and low-performance modes. The power consumption is evaluated using circuit simulations. The timing quality of the structures are evaluated by performing Monte Carlo simulations using NGSPICE. Each structure is simulated with 250 Monte Carlo simulations while applying wire width variations ($\pm 5\%$), voltage variations ($\pm 7.5\%$), channel length variations ($\pm 5\%$) and temperature variations ($\pm 15\%$) around the nominal values. The variations are spatially correlated and generated using a five-level quad-tree where equal magnitudes of variations are specified for each level of the constructed quad-tree [1, 9]. In each simulation, skew and transient time constraints are evaluated. If the chip meets timing constraints, the chip is classified as good. Otherwise, the chip is classified as defective. For all structures, no yield loss is obtained from the transition time constraints. The proposed setup overcomes the well-known deficiencies of the ISPD 2010 contest, which have been discussed at length in [3, 9], and [12]. Therefore, we have not reported extensive comparisons with the contest results.

The MRT design configurations are evaluated in Section 7.1. The performance of the MRT structures are evaluated in Section 7.2.

7.1 Evaluation of MRT Design Configurations

In this section, we evaluate the impact of the MRT design configurations. The selection of the number of top-level trees is evaluated in Section 7.1.1. The TRG guided the top-level tree construction is evaluated in Section 7.1.2. The use of topology reconfiguration is evaluated in Section 7.1.3. The delivery of the negative-edge of the clock signal is evaluated in Section 7.1.4.

7.1.1 Selection of the Number of Top-Level Trees. The selection of the number of top-level trees (N) is evaluated in Figure 6. The figure shows the performance of the MRT structure for different numbers of top-level trees on the circuits *dma* and *aes*. The performance is evaluated in terms of power and robustness to variations. The values presented in Figure 6 are normalized with respect to an MRT structure with $N = 1$, i.e., a clock network in the form of a clock tree. The robustness to variations is evaluated using D . The figure shows that D is improved when N is increased until a turning point from where the robustness is degraded. This is because each additional top-level tree improves the robustness to variations by placing clock sinks that are distant in the initial tree topology close in at least one of the subsequent top-level tree topology. However, with each top-level tree that is constructed, the clock network becomes larger and more vulnerable to variations.

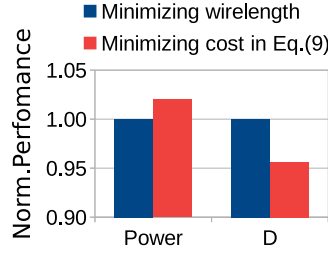


Fig. 7. Evaluation of guiding the top-level tree construction using different cost metrics. The evaluation is performed in terms of normalized power and normalized robustness (D).

Therefore, the robustness to variations is typically saturated when N is equal to three. Intuitively, the power consumption is increased with the number of top-level clock trees that are constructed, which can easily be observed in the figure.

Clearly, a fundamental problem for MRT structures becomes that of selecting N such that the minimum constraint on D is satisfied using the minimum amount of power consumption. This could potentially be solved using static timing analysis [12]. However, we consider this problem as being out of the scope of this article, as this article is focused on the concept and synthesis of MRT structures. Instead, we choose to evaluate the MRT structure with the two top-level trees on all circuits because it provides the best tradeoff between robustness and power consumption. Moreover, the MRT structures with more than two top-level trees introduce undesirable overheads in terms of both power consumption and runtime. Hence, the MRT structure with two top-level trees is selected to be the baseline for the proposed framework. In the remaining of this article, the MRT structure refers to the MRT structure that is obtained by constructing two top-level trees.

7.1.2 Evaluation of the TRG-Guided Top-Level Tree Construction. The TRG-guided top-level tree construction is evaluated in Figure 7. In the figure, we compare constructing the top-level trees with the objective of minimizing wirelength and using the proposed metric in Equation (9). The comparison is performed in terms of average power consumption and average robustness to variations (D) over all the benchmarks in Table 3.

Compared with minimizing wirelength, the figure shows that the proposed metric improves D with 4% for the MRT structures. The robustness improvements stem from the proposed cost metric that encourages clock sinks (with skew constraints) that are distant in the initial tree topology to be placed close in the subsequent tree topology. The robustness improvements come at the expense of a 2% overhead in power consumption. This is expected because subtrees that are distant in the initial tree topology are typically also distant spatially. Consequently, more wirelength is required to merge such subtrees, which intuitively increases power consumption.

Clearly, the proposed cost metric introduces a tradeoff between robustness and power consumption. We consider the small overheads in power consumption to be acceptable for the slight improvements in robustness. Consequently, we choose to guide the construction of the top-level trees using the proposed cost metric. Note that the tradeoff between robustness (D) and power consumption is regulated by the parameter α . In our implementation, we set α to prioritize topology distance over wirelength, i.e., α is set close to zero ($\alpha = 0.01$).

7.1.3 Evaluation of Topology Reconfiguration. In this section, we evaluate the effectiveness of saving power using topology reconfiguration combined with DVFS. In Figure 8(a), we compare the proposed technique with only applying DVFS in terms of supply voltage and clock network switching capacitance. The figure shows that the proposed technique results in 1% higher supply voltage

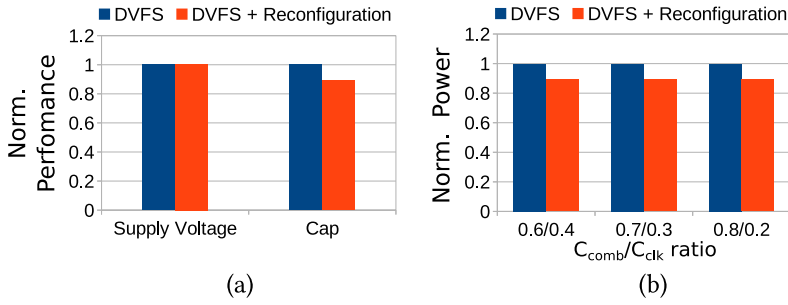


Fig. 8. Evaluation of DVFS vs reconfiguration combined with DVFS in terms of average (a) supply voltage and switching capacitance and (b) total circuit power for different ratios of C_{comb} and C_{clk} . The experimental results shown in the figure are the average values for the benchmarks in Table 3.

and 12% lower clock network switching capacitance. This is because the topology reconfiguration introduces small nominal skews, which reduces the available timing margins by 2%. The smaller timing margins translate to requiring a 1% higher supply voltage to be used. Compared with only applying DVFS, it is not surprising that the proposed technique results in only marginally higher supply voltage, as large timing margins are required to slightly reduce V_{DD} , when V_{DD} is close to V_t , as discussed in Section 5.3.

In Figure 8(b), we compare the proposed technique with DVFS in terms of total circuit power. The total circuit power is computed using Equation (8) for different ratios of switching capacitance between the combinational logic (C_{comb}) and the clock network (C_{clk}). The figure shows that the proposed technique reduces the total circuit power by about 11%. The power saving is relatively independent of the ratio between the capacitance of the combinational logic and the clock network. The power saving is obtained due to the lower clock network switching capacitance. Consequently, we conclude that topology reconfiguration combined with DVFS is advantageous to only applying DVFS. The only drawback of the reconfiguration is that a mode control signal is required to be delivered to the clock gates. Nevertheless, the overhead is limited because there is only one clock gate in the MRT structures. Moreover, the wirelength used to deliver the mode control signal does not contribute significantly to the dynamic power consumption. The results in the figure were obtained using α_{comb} and α_{clk} respectively set to 0.1 and 1. V_t for the technology is 0.4V [22].

7.1.4 Evaluation of the Negative-Edge of the Clock Signal. In this section, we evaluate the robustness in the delivery of the positive-edge and negative-edge of the clock signal. The comparison is performed in terms of D in Figure 9. The results shown in the figure are the average values obtained from the MRT structures for the benchmarks in Table 3.

The figure shows that MRT structure delivers both the positive-edge and negative-edge of the clock signal with higher robustness to variations than a Tree structure. However, the robustness improvements are higher for the positive-edge of the clock signal than for the negative-edge of the clock signal. As it is shown in Figure 9, the MRT structure delivers the positive-edge of the clock signal with 39% lower D while the negative-edge is only delivered with 24% lower D . The difference in the robustness improvements stem from that the OR-gates act as average operators and max operators for the positive-edge and negative-edge of the clock signal, respectively. This was discussed and analyzed in Section 5.2. Consequently, the MRT structures are ideal for circuits with only positive-edge triggered sequential elements. However, they can also be used for circuits with both positive-edge and negative-edge triggered sequential elements. Nevertheless, a slightly lower clock frequency will be required to be used.

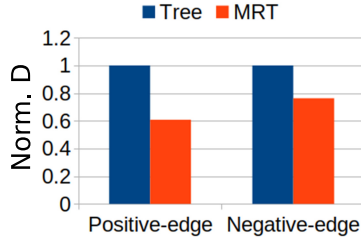


Fig. 9. Evaluation of the MRT structure for the negative-edge and the positive-edge of the clock signal.

7.2 Evaluation of MRT Structures

In Table 5, we evaluate the performance of MRT structures in both high-performance mode and low-performance mode. The structures are evaluated in terms of supply voltage, power consumption and the timing quality under variations. Recall that the MRT structures are obtained by performing the flow in Figure 3(a) using $N = 2$.

The supply voltage and power consumption of the structures are listed in the columns as “Supply Voltage” and “Power”, respectively. The timing quality of the structures are evaluated based on the timing constraints in Equations (1) and (2) in both modes of operation. The timing quality is reported as “Yes” or “No” in the column labeled “Timing Constraints Satisfied?”. The weighted clock power in Equation (7) is reported in the column labeled as ‘ P_T ’. The runtime of each structure is reported in minutes in the column labeled as ‘Runtime’. In the experimental setup, β and γ are set to be 0.8 and 0.2, respectively. The normalized results are shown in the bottom of the table and the best results in each column are shown in bold. The normalized results for the timing quality are reported as the total number of benchmarks that each structure meets the timing constraints.

Compared with the Tree structures, the Near-Tree-SS structures meet the timing constraints on three more benchmarks in both modes of operation. This is easy to understand because the Near-Tree-SS structures have alternative paths from clock source to clock sinks, which improves the robustness to variations. The Near-Tree-SS structures operate in 2% lower supply voltage in the high-performance mode, when compared to the Tree structures. The reason is that the Near-Tree-SS structures have more timing margins available due to the alternative paths from clock source to clock sinks, which enables the supply voltage to be scaled down slightly more when compared to the Tree structures. However, the supply voltage for the Tree structures and the Near-Tree-SS structures are similar in the low-performance mode because the timing constraints are significantly looser in the low-performance mode due to the lower operating frequency. The power consumption of the Near-Tree-SS structures is 37% higher than the Tree structures in both modes of operation, which results in a 37% higher P_T . The explanation for this is that the Near-Tree-SS structures have alternative paths from the clock source to the clock sinks, which introduce significant capacitive overheads. The runtime of the Near-Tree-SS structures are 10% shorter. The explanation is that clock tree optimization using SPICE simulations is more time consuming for Tree structures than for Near-Tree-SS structures.

The Near-Tree-SS structures meet the timing constraints on six benchmarks while the Near-Tree-OR structures meet the timing constraints on five benchmarks. This stems mainly from that OR-gates are slightly more vulnerable to variations than clock buffers. Recall that the only difference between the two structures is that OR-gates are used to join alternative redundant paths instead of sink-splitting. The Near-Tree-SS structures operate in 1% and 2% lower supply voltage than the Near-Tree-OR structures in the high-performance mode and the low-performance mode, respectively. This is because the Near-Tree-SS structures have more timing margins available than

Table 5. Evaluation of Structures in Terms of Power Consumption, Supply Voltage and Timing Quality

Benchmark (name)	Structure (name)	High Performance Mode		Low Performance Mode		Timing Constraints Satisfied? (Yes/No)	P_T in Equation (7) (mW)	Runtime (mins)
		Supply Voltage (V)	Power (mW)	Supply Voltage (V)	Power (mW)			
s1423	Tree [2, 5, 25]	1.02	35.97	0.61	2.25	No	29.23	2.1
	Near-tree-SS [9]	1.02	48.18	0.61	2.99	No	39.14	1.4
	Near-tree-OR	0.99	38.06	0.61	2.40	Yes	30.93	0.9
	MRT	0.99	36.27	0.61	2.06	Yes	29.43	0.9
usbf	Tree [2, 5, 25]	1.01	38.00	0.61	2.53	Yes	30.91	6.6
	Near-tree-SS [9]	0.98	55.66	0.61	3.65	Yes	45.25	5.4
	Near-tree-OR	0.98	48.06	0.62	3.24	Yes	39.10	3.6
	MRT	0.98	42.24	0.61	2.57	Yes	34.31	3.2
dma	Tree [2, 5, 25]	1.00	48.44	0.61	3.20	Yes	39.39	10.0
	Near-tree-SS [9]	0.98	71.02	0.61	4.80	Yes	57.78	7.3
	Near-tree-OR	0.99	70.68	0.64	4.76	Yes	57.49	7.5
	MRT	0.98	54.47	0.62	3.33	Yes	44.24	5.6
pci_bridge32	Tree [2, 5, 25]	1.01	72.97	0.61	4.84	Yes	59.35	13.2
	Near-tree-SS [9]	0.98	98.53	0.61	6.65	Yes	80.15	16.8
	Near-tree-OR	0.99	120.16	0.63	7.92	Yes	97.71	18.3
	MRT	0.99	83.54	0.62	5.04	Yes	67.84	10.1
ecg	Tree [2, 5, 25]	1.02	175.10	0.61	11.35	No	142.35	62.0
	Near-tree-SS [9]	0.99	227.66	0.61	15.06	Yes	185.14	86.5
	Near-tree-OR	1.02	257.65	0.63	15.65	No	209.25	85.7
	MRT	1.02	220.18	0.62	12.07	Yes	178.56	49.1
des	Tree [2, 5, 25]	1.02	176.36	0.61	11.43	No	143.37	133.0
	Near-tree-SS [9]	1.01	228.75	0.61	14.32	Yes	185.86	89.4
	Near-tree-OR	0.99	196.65	0.62	12.91	Yes	159.90	53.8
	MRT	0.99	177.76	0.61	11.08	Yes	144.42	53.5
aes	Tree [2, 5, 25]	1.02	334.18	0.62	21.62	No	271.67	300.0
	Near-tree-SS [9]	1.01	466.89	0.61	29.72	Yes	379.45	293.2
	Near-tree-OR	1.02	527.13	0.64	31.43	No	427.99	283.5
	MRT	1.01	386.55	0.62	21.01	Yes	313.44	213.0
Norm	Tree [2, 5, 25]	1.00	1.00	1.00	1.00	3	1.00	1.00
	Near-tree-SS [9]	0.98	1.37	1.00	1.37	6	1.37	0.90
	Near-tree-OR	0.99	1.37	1.02	1.35	5	1.37	0.80
	MRT	0.98	1.12	1.00	1.00	7	1.11	0.60

that of the Near-Tree-OR structures, which enables the supply voltage to be scaled down slightly more. The power consumption of the Near-Tree-OR structure is similar to the Near-Tree-SS structures in the high-performance mode. In the low-performance mode, the Near-Tree-OR structures have 2% lower power consumption than the Near-Tree-SS structures. The lower power consumption in the low-performance mode is a result of the Near-Tree-OR structures consume no short circuit power as there is only one gate driving each net. P_T is similar for both structures. The runtime of the Near-Tree-OR structures is 10% shorter than the Near-Tree-SS structures.

The MRT structures meet timing constraints on all benchmarks while the Near-Tree-OR structures fail to meet timing constraints on *aes* and *ecg*. The improvements in timing quality stem from that redundancy are inserted using top-level trees instead of using local-merges. The local-merges technique only considers if pairs of subtrees are sequentially related while the proposed TRG-guided top-level tree construction attempts to place every pair of clock sinks with timing constraints close in at least one tree topology. The supply voltage for the MRT structures is respectively 1% and 2% lower, as compared to the supply voltage for the Near-Tree-OR structures in the high-performance mode and the low-performance mode. The MRT structures have 25% and 35% lower power consumption than the Near-Tree-OR structures in the high-performance mode and the low-performance mode, respectively. Thus, the MRT structures have 26% lower P_T than the

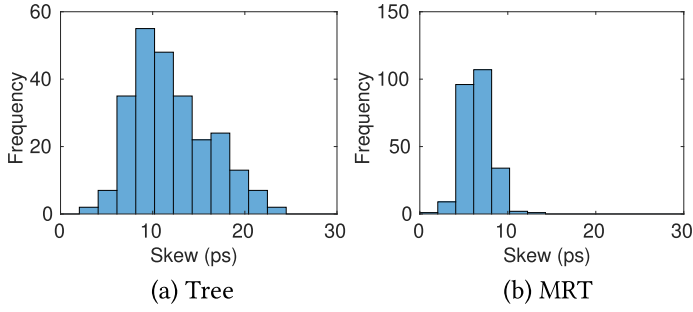


Fig. 10. Histogram of skews from Monte Carlo simulations of (a) Tree and (b) MRT structures on *usb_f*.

Near-Tree-OR structures. The improvement in P_T is a result of that the Near-Tree-OR structures have an excessive amount of redundant paths, which translates into larger clock networks with higher power consumption. Moreover, the MRT structures reconfigure the topology into a tree by turning off the redundant paths, which results in significant power savings in the low-performance mode. The MRT structures have similar power consumption with the Tree structures in the low-performance mode, which validates the effectiveness of the reconfiguration of the topology. The runtime of the MRT structures is 20% shorter than the runtime of the Near-Tree-OR structures.

To illustrate the robustness improvements of the MRT structures over the Tree structures, we show a histogram of the skew introduced by variations for both the Tree structure and the MRT structure on the circuit *usb_f* in Figure 10. As it can be observed, the MRT structure has a tighter skew distribution than the Tree structure, i.e., higher robustness to variations.

Intuitively, a clock tree is always the best clock network solution if it can satisfy the timing constraints without excessive guardbands. In this article, we focus on circuits where a higher robustness to variations is required. For such circuits, we have observed that the MRT structures are capable of providing higher robustness to variations when compared to the state-of-the-art Near-Tree-SS structures. Moreover, the MRT structures can provide significantly lower power consumption in both modes of operation. Consequently, we conclude that the proposed MRT structures are advantageous to the Near-Tree structures [9], which have been demonstrated to outperform clock trees with cross-links. Recall that clock meshes have 3X–5X high-power consumption than a clock tree. Therefore, the MRT structures are also advantageous to non-tree structures as the power overhead is as low as 12% in the high-performance mode. Furthermore, the topology of the MRT structure can be reconfigured to save additional power in the low-performance mode.

8 SUMMARY AND FUTURE WORK

In this article, we proposed the construction of clock networks with a mode reconfigurable topology for positive-edge triggered sequential elements. In high-performance modes, the proposed MRT structure is reconfigured into a near-tree to provide the required robustness to variations. In low-performance modes, power is saved by reconfiguring the MRT structure into a tree topology using a single clock gate. To the best of the authors' knowledge, this is the first clock network that can be reconfigured from a near-tree into a tree topology. Compared to the state-of-the-art near-tree structures, the experimental results demonstrate improvements in power consumption and robustness to variations. In the future, the authors plan to explore inserting OR-gates in multiple stages of the topology and integrating the use of useful skew. Moreover, we plan to target handling the delivery of the clock signal for both negative-edge and dual-edge triggered sequential elements and analyze the impact on the setup and hold times.

REFERENCES

- [1] A. Agarwal, D. Blaauw, and V. Zolotov. 2003. Statistical timing analysis for intra-die process variations with spatial correlations. In *Proceedings of the International Conference on Computer Aided Design (IEEE Cat. No.03CH37486)*. 900–907.
- [2] K. D. Boese and A. B. Kahng. 1992. Zero-skew clock routing trees with minimum wirelength. In *Proceedings of the IEEE International ASIC Conference and Exhibit*. 17–21.
- [3] Shashank Bujimalla and Cheng-Kok Koh. 2011. Synthesis of low power clock trees for handling power-supply variations. In *Proceedings of the International Symposium on Physical Design (ISPD'11)*. 37–44.
- [4] T.-H. Chao, Y.-C. Hsu, and J.-M. Ho. 1992. Zero skew clock net routing. In *Proceedings of the Design Automation Conference (DAC)*. 518–523.
- [5] Y. P. Chen and D. F. Wong. 1996. An algorithm for zero-skew clock tree routing with buffer insertion. In *Proceedings of the European Conference on Design and Test (EDTC'96)*. 230.
- [6] M. Edahiro. 1993. A clustering-based optimization algorithm in zero-skew routings. In *Proceedings of the Design Automation Conference (DAC)*. 612–616.
- [7] Rickard Ewetz, Shankarshana Janarthanan, and Cheng-Kok Koh. 2015. Construction of reconfigurable clock trees for MCM designs. In *Proceedings of the Design Automation Conference (DAC)*. 1–6.
- [8] Rickard Ewetz, Shankarshana Janarthanan, Cheng-Kok Koh, and Chuan Yean Tan. 2017. *Benchmark Circuits for Clock Scheduling and Synthesis*. (Version 3.0). Purdue University Research Repository. DOI: [10.4231/R7M32SSV](https://doi.org/10.4231/R7M32SSV)
- [9] Rickard Ewetz and Cheng-Kok Koh. 2015. Cost-effective robustness in clock networks using near-tree structures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 4 (2015), 515–528.
- [10] J.P. Fishburn. 1990. Clock skew optimization. *IEEE Trans. Comput.* 39, 7 (1990), 945–951.
- [11] S. Held, B. Korte, J. Massberg, M. Ringe, and J. Vygen. 2003. Clock scheduling and clocktree construction for high performance ASICs. In *Proceedings of the International Conference on Computer Aided Design (IEEE Cat. No.03CH37486) (ICCAD)*. 232–239.
- [12] Dong-Jin Lee and Igor L. Markov. 2011. Multilevel tree fusion for robust clock networks. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 632–639.
- [13] Jianchao Lu and Baris Taskin. 2009. Post-CTS clock skew scheduling with limited delay buffering. In *Proceedings of the International Midwest Symposium on Circuits and Systems (MWSCAS)*. 224–227.
- [14] Tarun Mittal and Cheng-Kok Koh. 2011. Cross link insertion for improving tolerance to variations in clock network synthesis. In *Proceedings of the International Symposium on Physical Design (ISPD)*. 29–36.
- [15] NGSPICE. [n. d.]. [Available Online] <http://ngspice.sourceforge.net/>.
- [16] A. Rajaram, Jiang Hu, and R. Mahapatra. 2004. Reducing clock skew variability via cross links. In *Proceedings of the Design Automation Conference, 2004. (DAC)*. 18–23.
- [17] Anand Rajaram, David Z. Pan, and Jiang Hu. 2005. Improved algorithms for link-based non-tree clock networks for skew variability reduction. In *Proceedings of the International Symposium on Physical Design (ISPD)*. 55–62.
- [18] Subhendu Roy, Pavlos M. Mattheakis, Laurent Masse-Navette, and David Z. Pan. 2015. Clock tree resynthesis for multi-corner multi-mode timing closure. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 4 (2015), 589–602.
- [19] Hyungjung Seo, Juyeon Kim, Minseok Kang, and Taewhan Kim. 2015. Synthesis for power-aware clock spines. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 126–131.
- [20] Xin-Wei Shih, Hsu-Chieh Lee, Kuan-Hsien Ho, and Yao-Wen Chang. 2010. High variation-tolerant obstacle-avoiding clock mesh synthesis with symmetrical driving trees. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 452–457.
- [21] V. Stojanovic, D. Markovic, B. Nikolic, M. A. Horowitz, and R. W. Brodersen. 2002. Energy-delay tradeoffs in combinational logic using gate sizing and supply voltage optimization. In *Proceedings of the European Solid-State Circuits Conference*. 211–214.
- [22] C. N. Sze. 2010. ISPD 2010 high performance clock network synthesis contest: Benchmark suite and results. In *Proceedings of the International Symposium on Physical Design (ISPD)*. 143.
- [23] C. N. Sze, Phillip Restle, Gi-Joon Nam, and Charles Alpert. 2009. ISPD 2009 clock network synthesis contest. In *Proceedings of the International Symposium on Physical Design (ISPD)*. 149–150.
- [24] Chung-wen Albert Tsao and Cheng-kok Koh. 2002. UST/DME: A clock tree router for general skew constraints. *ACM Trans. Des. Autom. Electron. Syst.* 7, 3 (2002), 359–379.
- [25] R.-S. Tsay. 1991. Exact zero skew. In *Proceedings of the International Conference on Computer-Aided Design Digest of Technical Papers*. 336–339.
- [26] Necati Uysal, Juan Ariel Cabrera, and Rickard Ewetz. 2020. Synthesis of clock networks with a mode reconfigurable topology and no short circuit current. In *Proceedings of the International Symposium on Physical Design (ISPD)*. 103–110.

- [27] Ganesh Venkataraman, Zhuo Feng, Jiang Hu, and Peng Li. 2006. Combinatorial algorithms for fast clock mesh optimization. In *Proceedings of the International Conference on Computer Aided Design (ICCAD)*. 563–567.
- [28] G. Venkataraman, N. Jayakumar, J. Hu, P. Li, Sunil Khatri, Anand Rajaram, P. McGuinness, and C. Alpert. 2005. Practical techniques to reduce skew and its variations in buffered clock networks. In *Proceedings of the International Conference on Computer-Aided Design. (ICCAD)*. 592–596.
- [29] Laung-Terng Wang, Yao-Wen Chang, and Kwang-Ting (Tim) Cheng. 2009. *Electronic Design Automation: Synthesis, Verification, and Test*. Morgan-Kaufmann Publishers, Inc., San Francisco, CA.
- [30] Linfu Xiao, Zigang Xiao, Zaichen Qian, Yan Jiang, Tao Huang, Haitong Tian, and Evangeline F. Y. Young. 2010. Local clock skew minimization using blockage-aware mixed tree-mesh clock network. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 458–462.

Received February 2021; revised October 2021; accepted November 2021