Multi-Corner Multi-Voltage Domain Clock Mesh Design

Can Sitik
Electrical and Computer Engineering
Drexel University
Philadelphia, PA, 19104 USA
E-mail: as3577@drexel.edu

Baris Taskin
Electrical and Computer Engineering
Drexel University
Philadelphia, PA, 19104 USA
E-mail: taskin@coe.drexel.edu

ABSTRACT

This paper introduces a novel multi-voltage domain clock mesh design methodology that is effective under multiple process corners. In multi-voltage designs, a single clock mesh that spans multiple voltage domains is infeasible due to the incompatibility of voltage levels of the clock drivers on the electrically-shorted mesh–each voltage domain requires a separate mesh. The skew among these isolated meshes need to be matched and a novel premesh tree synthesis is required to tolerate the impact of PVT variations exacerbated due to the separation of clock meshes for multiple voltage levels. The experiments performed with the largest three ISCAS'89 benchmark circuits operating at 500 MHz, 90 nm technology and 3 process corners show that: 1) The multi-voltage domain clock mesh can achieve up to 42% lower power on average with 39.04 ps skew, as low as $\approx$1.95% of the clock period, on average over a typical single voltage domain clock mesh, and 2) multi-corner optimized multi-voltage domain clock mesh can decrease the global skew of all corners by 190.42 ps on average over a multi-voltage domain clock mesh optimized for a single corner with a 15% degradation in power consumption necessary for variation-tolerance.

Categories and Subject Descriptors

B.7.1 [**Hardware**]: INTEGRATED CIRCUITS—*Types and Design Styles*

General Terms

Design

Keywords

VLSI, multi-corner clocking, multi-voltage design, low-power

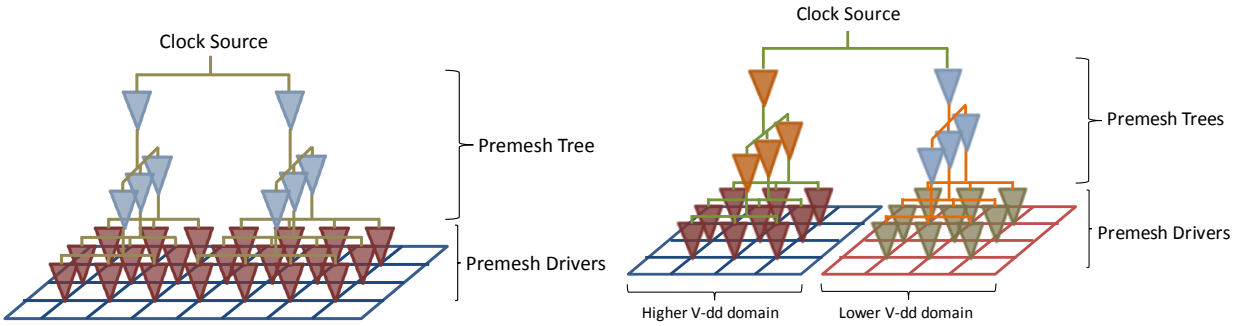
1. INTRODUCTION

In high performance IC clock design, the tradeoff between the performance and the power consumption is well-studied [1–9]. In ASIC design, clock tree topologies are preferred due to their low power profile with less wire demand and flexibility to be combined with low power techniques such as multi-voltage domain design [10]. On the other hand, in high-end microprocessor design, clock topologies with redundancies are preferred due to their tolerances against process variations with the cost of extra wire capacitance. In the clock tree topologies with spines and crosslinks [1, 9], selected branches are shorted to decrease the skew mismatch in the presence of variations, whereas in clock mesh topologies, every branch is shorted in the design to provide a global improvement of the skew [2–4, 6–8]. Literature on clock meshes focuses on design methodologies for mesh reduction and optimization, and ignores multi-corner analysis, as the ubiquitous shorting of branches provides remarkable robustness under PVT variations. However, this assumption only holds true for a traditional, single-Vdd clock mesh. When a system-on-chip style multi-Vdd single-clock domain design is considered, or for a design employing the low power technique of multi-voltage design, a clock network built with the mesh-topology becomes uncharacteristically susceptible to PVT variations. This susceptibility emerges as a multi-voltage clock mesh is constructed with disjoint premesh trees and meshes, each synchronizing an individual voltage domain.

In this work, a novel multi-corner multi-voltage clock mesh design methodology is presented. Multi-voltage clock mesh design is not a straight-forward process: First, a single clock mesh is not feasible as its electrically-shorted mesh wires cannot drive the voltage sinks operating at different voltage levels. Separate meshes are needed for each domain. Second, the skew among these domains must be balanced, which is a challenge that arises due to the isolation of the premesh trees among the voltage domains. Third, the skew introduced by the variation must be analysed and tuned considering multiple process corners, as isolated premesh trees in different voltage domains have different tolerances against the PVT variations. A previous work in [11] addressed the first two objectives but fails to consider the third objective of variation-tolerance. Variation-awareness of the multi-voltage clock mesh topology necessitates a novel design scheme, which is proposed in this paper to alleviate the impacts on timing variation, which are further exacerbated due to the presence of multiple voltage domains. Hence, a novel methodology that addresses all three of these objectives concurrently is necessary to synthesize a variation-tolerant, power-efficient multi-voltage domain clock network.

The experiments performed on the largest circuits of ISCAS'89 with 90 nm models at 500 MHz highlights two important benefits of multi-corner multi-voltage domain clock mesh design: 1) Proposed multi-corner multi-voltage domain clock mesh can achieve up to 42% lower power at three process corners with a 39.04 ps skew on average, that resides in the 2% budget of the clock period, 2) proposed method can achieve 190.42 ps lower skew in average and a much tighter slew compared to optimizing multi-voltage domain clock mesh design at only one process corner, with a 15% degrada-

GLSVLSI'13, May 2–3, 2013, Paris, France.



(a) Typical Single-Voltage Domain Clock Mesh Topology.

(b) Proposed Multi-Voltage Domain Clock Mesh Topology.

Figure 1: Clock Mesh Topologies

tion in the power consumption at that corner (but functional as the skew is corrected not to be larger than an allowable skew budget).

The rest of the paper is organized as follows. In Section 2, the preliminaries of the clock mesh design and the challenges of multi-voltage clocking are briefed. In Section 3, the proposed multi-corner multi-voltage domain clock mesh design methodology is explained in detail. In Section 4, experimental results are presented. The paper is finalized with concluding remarks in Section 5.

2. PRELIMINARIES AND BACKGROUND

The clock mesh topologies and their challenges are described in Section 2.1. The effect of multiple voltage-domains on the clock buffers is discussed in Section 2.2.

2.1 Clock Mesh Design

There are two major design specifications of the clock mesh design shown in Figure 1. First is the power consumption, introduced by the RC effect of the redundant mesh wires, and the second is the clock skew, introduced by the variations and geometric mismatches.

2.1.1 Power Dissipation in Clock Mesh

The power consumption of a clock mesh can be approximated as follows:

$$P_{total} \approx \alpha c_{total} f V_{dd}^2 \tag{1}$$

where α is the switching factor, f is the operating frequency and V_{dd} is the supply voltage. The switching capacitance c_{total} is:

$$c_{total} \approx c_{mesh} + c_{stub} + c_{pmt} + \sum_{\forall i} c_i \tag{2}$$

where the last term stands for the total capacitance of the sink registers, and c_{mesh}, c_{stub} and c_{pmt} are the total capacitance of the mesh wires, stub wires and the premesh tree, respectively.

Under the same switching activity α and the same frequency f, there are two ways to decrease the power dissipation of the synthesized clock mesh network. First is to decrease the total capacitance, for which [2, 6] propose mesh wire reduction, [3] proposes stub wire reduction and [4, 7] propose both. Second is the supply voltage reduction, which can be realized using multi-voltage domains as described in this paper. Although multi-voltage mesh has unique challenges, it is more promising than capacitance reduction because 1) most of the switching capacitance is at the clock sinks, and thus, mesh or stub wire length reduction does not reduce the dominant switching capacitance component, and 2) supply voltage reduction scales power quadratically, rather than linearly. Besides, voltage domains are not proposed to exclusively reduce the clock power but as a necessity due to the voltage domains being created for the logic (e.g. an SoC or multi-Vdd design) and not for the clock itself. It is an added incentive that these multiple voltage domains can help reduce the power dissipation of a clock mesh. The culprit is the undesired increase in clock skew between the meshes on each individual voltage domain, which is addressed in this paper.

2.1.2 Skew in Clock Mesh

The global clock skew in the mesh is estimated as:

$$t_{skew} = t_{skew}^{pmt} + t_{skew}^{mesh} + t_{skew}^{stub} \tag{3}$$

where t_{skew}^{pmt}, t_{skew}^{mesh} and t_{skew}^{stub} are the skews introduced by the difference between the maximum and the minimum delays on the premesh tree of the mesh, the difference between the maximum and the minimum delays on the mesh from a premesh driver to a stub wire connection point on the mesh and the difference between the maximum and the minimum delays from a connecting point of a stub wire to a sink register, respectively. The skew introduced by t_{skew}^{mesh} and t_{skew}^{stub} can be reduced by using dense meshes in each domain, however, decreasing the skew introduced by t_{skew}^{pmt} is a challenge for multi-voltage meshes as the premesh trees depicted in Figure 1(b) are isolated.

2.2 Insertion Delay and Its Variation in Multi-Voltage Designs

Any multi-voltage clocking algorithm has the following challenges. First, the gate delay changes with the switching of the supply voltage between voltage domains. Second, the variation in the gate delay is not the same for the identical gates with different supply voltages. Clock skew is defined as the difference between the maximum and the minimum insertion delays, which may fall on different branches within separate voltage domains. To that end, it is proposed in this paper that the insertion delay profile of a clock mesh network be carefully investigated to optimize the skew, as this investigation reveals how to design these premesh tree branches within separate voltage domains in the presence of PVT variations of multiple process corners. In order to analyse how the insertion delay varies depending on the applied voltage, a motivational example is presented with 2 voltage domains, each with a simple 2-level clock tree with 16 sinks, as depicted in Figure 2.

In this motivational example, one domain is supplied with 1.2V and the other is with 0.8V, and SPICE models of *SAED90nm EDK library* of *Synopsys* are used [12]. This simple example is synthesized by *IC Compiler* of *Synopsys* using the same size of clock buffers. This circuit is simulated in *CustomSim XA* simulator of

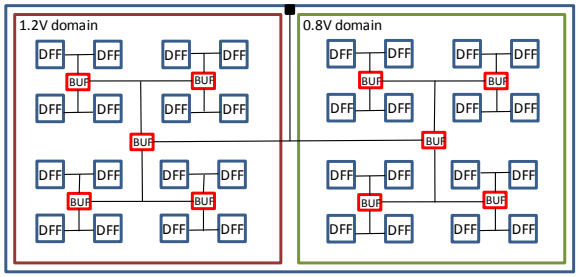


Figure 2: Simple 2-level Clock Tree with 16 Sinks

Table 1: Insertion Delay Profile of the Motivational Clock Tree Example at 2 Domains in Figure 2, the Max and the Min Insertion Delays that Define the Global Skew are Marked with Bold

	Best (ps)	Nom (ps)	Worst (ps)
Min ins delay in 0.8V D.	104.73	157.77	301.71
Max ins delay in 0.8V D.	**106.64**	**161.50**	**309.57**
Inc. compared to Best	-	≈ 55	≈ 203
Min ins delay in 1.2V D.	**101.36**	**130.71**	**223.46**
Max ins delay in 1.2V D.	101.65	131.29	224.65
Inc. compared to Best	-	≈ 30	≈ 122
Global Skew	5.28	30.79	86.11

Synopsys at 500 MHz, and the insertion delay is measured in 3 different process corners: The best, nominal and the worst cases of the process. The insertion delay profile monitored is shown in Table 1.

From Table 1, it is observed that the insertion delay variation is higher in the 0.8V domain which also increases the skew switching from the best case to the worst case: Maximum insertion delay increases by ≈203 to 309.57 ps, whereas the minimum insertion delay only increases by ≈122 to 223.46 ps. In this undesirable case, the maximum insertion delay increases the largest, as it is affected by PVT variations the most going from the best to worst case. The minimum insertion delay, on the other hand, increases the least, which is also undesirable. It is desirable, instead, that the maximum insertion delay path be delayed the least and the minimum insertion delay be delayed the most, to contain the skew under PVT variations going from the best to the worst case. The basis for the novel premesh tree methodology in this paper is that the maximum insertion delay path should be placed in the higher voltage domain at the best case *on purpose*, as paths in the higher voltage domain get affected less by PVT variations, and the skew budget is lower at all corners. To demonstrate this postulate, the insertion delay of the 1.2V domain is increased gradually by upsizing the first level of the two-level motivational clock tree first, and then both levels together to observe these two desirable delay profiles. It is expected that the insertion delay in the 1.2V domain increases by a delay offset in all PVT corners, which may increase the skew in the best case corner, but decrease the skew in the worst case, which means a decrease in the global skew.

From Table 2, it is seen that the global skew is decreased when the insertion delay is gradually increased in the 1.2V domain by upsizing the clock buffers. It is clear that, a novel clock network synthesis methodology should control the insertion delay of the different domains by either adding/removing buffers or upsizing/downsizing over a naive, single-corner optimized tree. Notice how the minimum and the maximum insertion delays (marked in bold) are consistently in the same voltage domain in all three corners in the undesired case, shown in Table 1. In the desired case, shown in Table 2, the delays vary between paths in either voltage domains for PVT corners, the skew being affected less from PVT variations. Although the skew is increased in the best corner (from 5.28 ps to 25.4 ps), global skew is defined as the worst case of all corners: Thus, global skew is improved from 86.11 ps to 39.47 ps, when the insertion delay is shifted by an upsizing. This example concludes that optimizing the skew at all three corners simultaneously requires an early arrival of the clock signal in the lower voltage domain at the best case so as to have a balanced skew at the worst case, and for the skew budget to be lower. Multi-corner optimization is available only in single-voltage mesh synthesis in the standard EDA tools. Therefore, the multi-corner optimized premesh tree synthesis methodology for the multi-voltage clock mesh designs developed in this paper fills this gap.

3. METHODOLOGY

A new methodology to synthesize a multi-corner mesh-based clock distribution network in a multi-voltage design flow is proposed. By considering multiple process corners at the same time, variation-tolerance is targeted. The steps of the proposed methodology are as follows: Given an initial placement and the mesh size,

1. Premesh drivers are placed at each intersection, and sized considering the slew constraints (Algorithm 1).

2. Premesh tree is synthesized iteratively considering all process corners to minimize the global skew (Algorithm 2).

For the multi-voltage clock mesh, local meshes are placed for each domain with their own premesh trees. The roots of the premesh trees of each domain can be connected to a master root through level shifters, however, these level shifters are not strictly necessary. The virtual root of each domain can function as a level shifter, instead, if the source voltage swing is higher than that of the supply voltage at the root. In order to eliminate the extra power cost, the level shifters are not used in this work, as shown in Figure 1(b). The highest voltage in the design is assumed to be the voltage swing of the clock source connection of the multi-voltage clock mesh design without loss of generality of the presented work. Buffer driver selection and sizing, and the premesh tree synthesis procedures are presented in Section 3.1 and Section 3.2, respectively. Section 3.1 is a simple algorithm that does not define the novelty of this work but complements the design flow. Note that, the proposed methodology is independent of the premesh driver sizing algorithm; any alternative, more (or less) sophisticated method for premesh driver sizing can be seamlessly integrated, which demonstrates the practicality of the proposed method for automation purposes. Furthermore, the algorithms presented in this work are designed for any number of voltage domains, however it is important to note that the number of voltage domains is 2 in the selected technology library, which is also the same library and setup in [11], against which the results are compared.

3.1 Premesh Driver Selection

In this step, a simple heuristic is used to select the premesh driver buffers. First, maximum size buffers in the library are placed at each intersection point of the clock mesh, then, the buffer sizes are decreased iteratively until the maximum slew exceeds the slew constraint at any of the corners. Note that, this algorithm is very similar to the premesh driver selection algorithm in [11], with one

Table 2: Improved Skew Numbers with a Delay Offset in the Insertion Delay of the 1.2V Domain

	Only first level upsized by one			Both two levels upsized by one		
	best (ps)	nom (ps)	worst (ps)	best (ps)	nom (ps)	worst (ps)
Min insertion delay in 0.8V domain	**104.64**	157.52	301.12	**104.64**	**157.51**	301.16
Max insertion delay in 0.8V domain	106.53	**161.29**	**308.93**	106.56	161.28	**308.98**
Min insertion delay in 1.2V domain	115.23	**145.07**	**239.61**	129.89	164.77	**269.51**
Max insertion delay in 1.2V domain	**115.53**	145.62	240.77	**130.04**	**165.12**	270.21
Global Skew	10.89	16.22	68.16	25.4	7.61	39.47

Algorithm 1 Premesh Driver Selection

Input: Mesh size for each voltage domain i, buffer library, slew constraint $slew_{const}$
Output: The sizes of each premesh driver buffers
  for Each voltage domain i **do**
    Place maximum size buffer at each intersection
    Calculate $slew^i_{max}$
    while $slew^i_{max} < slew_{const}$ **do**
      Replace all buffers with the next smaller buffer cell in the library
      Update $slew^i_{max}$
    end while
  end for
  Return the selected driver for each domain i

definite difference: [11] starts with minimum size buffers and performs upsizing whereas in this work, maximum size buffers are selected at the beginning and downsizing is performed iteratively. This approach decreases the number of iterations as the slew constraints are much tighter in this work compared to [11] and larger size buffers are needed. Furthermore, this similarity makes fair comparisons to [11], and is necessary to highlight the elegance of multi-corner optimized premesh trees (Section 3.2) over single-corner optimized premesh trees. The slew constraint $slew_{const}$ is set to its typical value of 5% of the clock period, although it can be set to another value depending on the performance requirements. Note that this constraint may be degraded by the synthesized premesh tree in the following procedure, yet the slew target needs to be selected in this stage as it is used as a guide that drives the premesh tree synthesis stage. In the premesh driver selection procedure, all the buffers are sized down at the same time with an identical scale for the sake of simplicity. This procedure is shown in Algorithm 1.

3.2 Multi-Corner Premesh Tree Synthesis

Premesh tree synthesis is the novel step of this proposed work that synthesizes a premesh tree to optimize the global skew considering all process corners simultaneously. This stage also considers the slew at the sinks, not to degrade the slew more than an allowed margin within the slew budget, as the maximum slew (Section 3.1) of the design is not completely independent from the premesh tree.

It is important to note the significance of the proposed skew-and-slew budgeted multi-corner, multi-voltage clock mesh design methodology within the current state of the art clock mesh design automation: In the multi-voltage design flow, standard EDA tools cannot match the insertion delays of the premesh trees driving clock meshes that belong to different voltage domains even in a single corner [11]. In this work, an iterative algorithm is proposed to accomplish this task at all process corners. In the algorithm, the premesh tree is synthesized iteratively, until the lowest overall (e.g. global) skew at all corners is achieved. This procedure is shown in Algorithm 2.

Algorithm 2 Multi-Corner Multi-Voltage Domain Premesh Tree Synthesis

Input: Premesh drivers for each domain, assigned voltage for each domain, buffer library for each domain i and each corner c, slew constraint $slew_{const}$
Output: Premesh tree for each domain i
  Set N = 1
  while $slew_{max} > slew_{const} + slew_{margin}$ **do**
    Place $N \times N$ max size buffers at the (n-1)st level of each domain i
    $N = N + 1$
  end while
  if *max insertion delay is in the same domain for all cases* **then**
    while *max insertion delay is still at the same domain j* **do**
      Add a max size buffer of that voltage domains other than j
    end while
  end if
  $curr_{skew} = \max_{\forall c}(max_{\forall i}(delay^i_{max})$ - $(\min_{\forall i}(delay^i_{min}))$
  $prev_{skew} = T$
  while $k_j < num-of-levels-in-j$ **do**
    Downsize the k-th level of the max insertion delay domain j
    $prev_{skew} = curr_{skew}$
    $curr_{skew} = \max_{\forall c}(max_{\forall i}(delay^i_{max})$ - $(\min_{\forall i}(delay^i_{min}))$
    if $curr_{skew} > prev_{skew}$ **then**
      Undo that step, mark k-th level as done
      $k_j = k_j + 1$
    end if
  end while
  while $curr_{skew} > prev_{skew}$ **do**
    Add parallel buffers at the max insertion delay domain at the first level
    $prev_{skew} = curr_{skew}$
    $curr_{skew} = \max_{\forall c}(max_{\forall i}(delay^i_{max})$ - $(\min_{\forall i}(delay^i_{min}))$
  end while

In this algorithm, first the (n-1)st level of the premesh tree is synthesized with the maximum size buffers in an N×N topology. The iterations starts (generically) with N=1, and it is increased by 1 (also generically) at each iteration until the slew constraint with the degradation margin is met at all corners. Then, the insertion delays are measured to see if the maximum insertion delay path is in the same domain for all cases. If so, a maximum size buffer is added in the next level of each domain other than the maximum insertion delay domain because it is an undesired case, as explained in Section 2.2. The purpose of selecting maximum size buffers in this stage is the fact that having less number of levels in the premesh tree with larger size buffers is more variation-tolerant than having more number of levels with smaller buffers. Buffer adding

Table 3: Created Benchmark Circuits

	0.8V Domain	1.2V Domain
Benchmark 1	s35932	s38417
Benchmark 2	s38417	s35932
Benchmark 3	s35932	s38584
Benchmark 4	s38584	s35932

stage ends when the maximum insertion delay paths are no longer encountered in the same domain for different corners. The buffer addition stage is succeeded by the buffer sizing stage. In the buffer sizing stage, the buffers are downsized starting from the first level of the domain whose maximum insertion delay path sets the global skew. If downsizing improves the skew, that step is performed. Otherwise, that step is undone and that level is marked as "done". Downsizing continues with the next maximum insertion delay domain and finalizes when all levels of all domains are marked as "done". After that, pruning stage starts. In this last stage, the number of levels or the sizes of the buffers are unchanged, but parallel buffers of the same sizes are added at the first level of the maximum insertion delay domain iteratively as long as the global skew is improved. The algorithm is an heuristic and does not guarantee an optimal solution. However, the algorithm is scalable with a complexity of O(n) for n voltage domains and m process corners. The complexity is linear because the algorithm visits every domain constant number of times, at each stage of the algorithm and simulates for the best and the worst cases only. Certain simplifications are done, such as choosing identical size buffers, which degrade the quality of the solution, but are sufficient to demonstrate the novelty of this work.

4. EXPERIMENTAL RESULTS

The proposed methodology is implemented with Tcl in order to inter-operate with standard EDA tools and tested on the benchmark circuits created with two voltage domains. The selected *SAED 90nm EDK Library* [12] has two voltage levels, a high voltage at 1.2V and a low voltage at 0.8V. For testing purposes, 4 benchmark circuits are created by placing different combinations of the three largest circuits of ISCAS'89 benchmarks, s35932, s38417 and s38584 within two voltage domains. These combinations are shown in Table 3.

The RTL level designs are synthesized using *Design Compiler* of *Synopsys* and the placement of the circuits are performed using *IC Compiler* of *Synopsys*. The skew and the power analysis are performed using *CustomSim XA* Simulator of *Synopsys* at the SPICE accuracy level with the SPICE models. Due to the tight slack constraints at the domain that operates at 0.8V, 500 MHz is selected as the operating frequency. In order to verify the variation-tolerance, all the tests are performed at 3 different process corners:

1. Best Corner (BC): V=V_{dd}+10%, T=-40°C, fast transistors
2. Nominal Corner (NC): V=V_{dd}, T=25°C, typical transistors
3. Worst Corner (WC): V=V_{dd}-10%, T=125°C, slow transistors

In order to compare the quality of results with the standard single-voltage mesh and [11], the following procedure is performed: First, the same circuits are synthesized with a single-voltage domain mesh whose premesh drivers are sized using the proposed method given in Section 3.1. For consistency, the premesh trees are synthesized using *IC Compiler* in the multi-corner mode. Then, a custom implementation of the multi-voltage domain clock mesh design algorithm in [11] is applied on the same circuits to compare to the

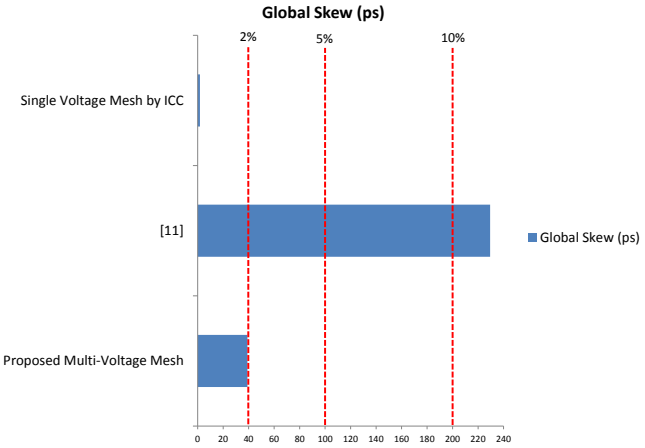


Figure 3: Skew Comparison vs. Typical Skew Budgets

proposed work. The maximum slew $slew_{const}$ is selected as 100 ps, with a degradation margin allowance in premesh tree synthesis of 20% totalling to a slew budget of 120 ps, as explained in Section 3.1 and Section 3.2. The mesh size is selected as 10×10, similar to [11], to have a fair comparison.

The proposed methodologies are applied on the 4 benchmarks in Table 3. The premesh buffer sizing (Section 1) takes 2 iterations, and the premesh tree synthesis stage (Section 3.2) takes 9 iterations at most to converge. The experimental results are shown in Table 4 and Table 5. Recall that the contemporary practical objective in clock distribution network design is to optimize power within a given skew/slew bound. It is visible from Table 4, that a single voltage mesh is the best in terms of clock skew, as the single voltage mesh (famously) has a very high variation-tolerance. The important contribution of this paper, is that, when multi-voltage designs are implemented, the clock mesh can preserve its variation-tolerance while reducing the power dissipation through multi-Vdd design. The proposed methodology can achieve up to 42% less power consumption in the clock network compared to a single voltage mesh. The skew increases from 1.73 ps to 39.04 ps on average, shown in Table 5, yet still as low as ≈1.95% of the clock period so well within the skew budget. In comparison, note that the previous single-corner work in [11] does not satisfy the 5% (or 10%) skew budget at 229.46 ps, ≈11.5% of the clock period, on average, highlighted in Figure 3. It is concluded that [11] is useful in demonstrating the feasibility of a multi-voltage mesh but the process was completed at a single corner. That methodology in [11] does not generate functional meshes at the multi-corner analysis. Similar to multi-corner multi-mode (MCMM) versions of design automation routines, the multi-corner multi-voltage mesh design methodology proposed in this paper employs novel and different algorithms to achieve demonstrated results.

It is reported in Table 4 that the power dissipation in the multi-voltage clock network is much lower compared to a single voltage clock mesh. The total power savings on the multi-voltage circuit will be even higher considering the additional savings on the logic, which are not reported here (as total power savings depend on the particular voltage domain partitioning technique and may sway the results). There is a 15% degradation in power consumption in the nominal case over the single-corner design in [11], as optimization at multiple corners is considered. The 15% power degradation, thus, is not a penalty but a necessity to achieve a slew and a skew constrained design. The slew constraint in [11] is 120 ps in

Table 4: Power Comparison over Single-Voltage Domain Mesh and [11] at 3 corners, BC, NC and WC

Circuits	SV Mesh Synthesized by *ICC*			[11]			Proposed Work		
	BC(mW)	NC(mW)	WC(mW)	BC(mW)	NC(mW)	WC(mW)	BC(mW)	NC(mW)	WC(mW)
Benchmark 1	467.64	112.69	201.89	272.56	66.52	137.90	304.19	82.97	132.72
Benchmark 2	502.04	128.33	220.41	219.09	80.28	148.54	254.88	95.96	140.25
Benchmark 3	385.19	93.17	178.90	197.55	48.34	104.18	231.41	64.51	102.70
Benchmark 4	434.58	106.73	184.59	212.60	72.29	145.62	239.84	88.26	132.65
	Avg. % Improvement			49%	40%	32%	42%	25%	35%

Table 5: Skew Comparison over Single-Voltage Domain Mesh and [11] at 3 corners, BC, NC and WC, Overall Skew is Bold

Circuits	SV Mesh Synthesized by *ICC*			[11]			Proposed Work		
	BC (ps)	NC (ps)	WC (ps)	BC (ps)	NC (ps)	WC (ps)	BC (ps)	NC (ps)	WC (ps)
Benchmark 1	**1.30**	0.97	0.63	**228.03**	21.06	177.31	**40.17**	20.16	28.80
Benchmark 2	1.54	2.65	**4.03**	**227.41**	21.64	178.73	**38.88**	18.54	30.94
Benchmark 3	**2.53**	1.90	1.19	**229.58**	20.60	179.16	30.26	9.99	**39.11**
Benchmark 4	**1.56**	1.25	0.76	**232.82**	16.42	170.09	30.26	10.14	**38.00**
Avg. Skew			1.73			229.46			39.04
	Degradation					-227.73			-37.31

the nominal case, but it is observed to be 240 ps in the worst case, which requires an extravagant 12% slew budget. In comparison, the worst case slew budget in this paper is 120 ps, which is satisfied at 108 ps, 5.4% of the period, in the worst case. For variation-awareness, and the saved timing slack, the power consumption is degraded by 15% in the nominal case. This degradation is 7% in the best case corner, and there is a 3% improvement in the worst case corner, which highlights the multi-corner optimization objective of the proposed work.

5. CONCLUSIONS

A new methodology is proposed to synthesize variation-tolerant, multi-voltage, single clock domain clock meshes for high performance ICs. The proposed method enables a multi-corner clock mesh synthesis on designs with multiple voltage domains. This is a substantial novelty in demonstrating the problem with multi-voltage designs and their synchronization with clock meshes and developing design automation routines for the methodology whose unavailability in industrial tools also demonstrated in experiments. Slew-and-skew budgeted premesh clock tree design at all process-voltage-temperature (PVT) corners is proposed for high practicality. In this work, a simple algorithm is used to size the premesh drivers to highlight the elegance of multi-corner optimized multi-voltage domain clock meshes. Therefore, this methodology can easily be combined with existing buffer sizing algorithms for improved results and a wide practical applicability. If more number of voltage levels is available, the proposed methodology can achieve a wide range in the skew vs. power curve by exploiting different voltage levels for different performance requirements.

6. REFERENCES

[1] A. Rajaram and D. Z. Pan, "Variation tolerant buffered clock network synthesis with cross links," in *Proceedings of the International Symposium on Physical Design (ISPD)*, 2006, pp. 157–164.

[2] A. Abdelhadi, R. Ginosar, A. Kolodny, and E. G. Friedman, "Timing-driven variation-aware nonuniform clock mesh synthesis," in *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI)*, May 2010, pp. 15–20.

[3] M. R. Guthaus, G. Wilke, and R. Reis, "Non-uniform clock mesh optimization with linear programming buffer insertion," in *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, June 2010, pp. 74–79.

[4] Y. Teng and B. Taskin, "Clock mesh synthesis method using the Earth Mover's Distance under transformations," in *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, Oct. 2012, pp. 121–126.

[5] J. Lu, Y. Teng, and B. Taskin, "A reconfigurable clock polarity assignment flow for clock gated designs," *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems*, vol. 2, no. 6, pp. 1002–1011, June 2012.

[6] G. Venkataraman, Z. Feng, J. Hu, and P. Li, "Combinatorial algorithms for fast clock mesh optimization," *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, vol. 18, no. 1, pp. 131–141, January 2010.

[7] J. Lu, X. Mao, and B. Taskin, "Integrated clock mesh synthesis with incremental register placement," *IEEE Transactions on Computer-Aided Design (TCAD)*, vol. 31, no. 2, pp. 217–227, Feb. 2012.

[8] N. Kurd, J. Barkarullah, R. Dizon, T. Fletcher, and P. Madland, "A multigigahertz clocking scheme for the Pentium 4 microprocessor," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 36, no. 11, pp. 1647–1653, Nov. 2001.

[9] G. Shamanna, N. Kurd, J. Douglas, and M. Morrise, "Scalable, sub-1w, sub-10ps clock skew, global clock distribution architecture for Intel Core i7/i5/i3 microprocessors," in *Proceedings of the IEEE Symposium on VLSI Circuits (VLSIC)*, June 2010, pp. 83–84.

[10] C.-C. Tsai, T.-H. Lin, S.-H. Tsai, and H.-M. Chen, "Clock planning for multi-voltage and multi-mode designs," in *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*, March 2011, pp. 1–5.

[11] C. Sitik and B. Taskin, "Multi-voltage domain clock mesh design," in *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, Oct. 2012, pp. 201–206.

[12] *Synopsys 90nm Generic Library*, Synopsys Inc., 2009.