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 investigates the effectiveness of a multi-voltage clock network design that is built using the mesh topology. Unlike a clock tree, 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. These disjoint meshes need to be matched in clock skew between the domains. In addition, the additional power dissipation of the level shifters in the logic needs to be compared against the power savings of multi-voltage domain implementation. The case study performed with the largest ISCAS'89 benchmark circuits operating at 500 MHz, 90 nm technology concludes two important results that highlight the benefits of multi-voltage clock mesh design: 1) The multi-voltage domain clock mesh can achieve 37.14% lower power with a 9 ps increase in clock skew over the single-voltage domain clock mesh, and 2) The multi-voltage domain clock mesh achieves 66 ps less skew with a 20.92% increase in power dissipation over a multi-voltage domain clock tree.**

I. Introduction

In high performance IC design, the tradeoff between the maximum clock skew and the power consumption is well-studied [1–10]. Clock tree topologies are preferred in low-power ASIC design due to their low power consumption. Clock trees consume less metal wire for routing, and can be easily combined with multi-voltage domain low-power design techniques [11, 12]. Multi-voltage clocking with a tree-topology distribution network is a popular technique used for low-power design methodology with sophisticated automation techniques existent in industrial tool flows. Multi-voltage clock trees are designed to deliver clock signal with local trees in each voltage domain and these trees are connected at the upper level through level shifters [11]. On the other hand, clock structures with redundancies are preferred in high performance microprocessor design due to their low clock skew and tolerance to clock skew variations, despite their higher power consumption introduced by the extra wire capacitance. Among these clock structures, clock tree with spines or cross links [1–3] adds shortcuts between nodes to reduce the clock skew variations between selected branches while clock mesh [4–10] adds more redundancy by adding horizontal and vertical metal wires to short every clock branch to minimize the clock skew globally. What is missing in the literature and in automation flows is the combination of clock structures with redundancies and the multi-voltage clocking techniques. This work is presented to this end, exclusively targeting clock meshes.

In this work, a multi-voltage clock mesh design methodology is presented. Multi-voltage clock mesh design is not a straight-forward process, due to its unique challenges. 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 among the domains. Third, the level shifter overhead in the clock network of multiple clock domains must be considered at the design stage, similar to the consideration of the level shifter overhead in the logic network before multi-voltage design partitioning is performed.

The results of the tests performed on a case study with 2 voltage domains at 500 MHz show that:

1) The power dissipation of the multi-voltage clock mesh network decreases by 37.14% with a 9 ps increase in the skew over the single-voltage clock mesh design,
2) The power dissipation of the multi-voltage clock mesh network increases by 20.92% with a 66 ps decrease in the skew over the multi-voltage clock tree network,

with the proposed multi-voltage domain clock mesh.

The rest of the paper is organized as follows. In Section II, the preliminaries and the background for the clock mesh design and the multi-voltage clocking techniques are briefed. In Section III, the proposed multi-voltage clock mesh design methodology is explained in detail. In Section IV, a case study is presented. The paper is finalized with concluding remarks in Section V.

II. Preliminaries and Background

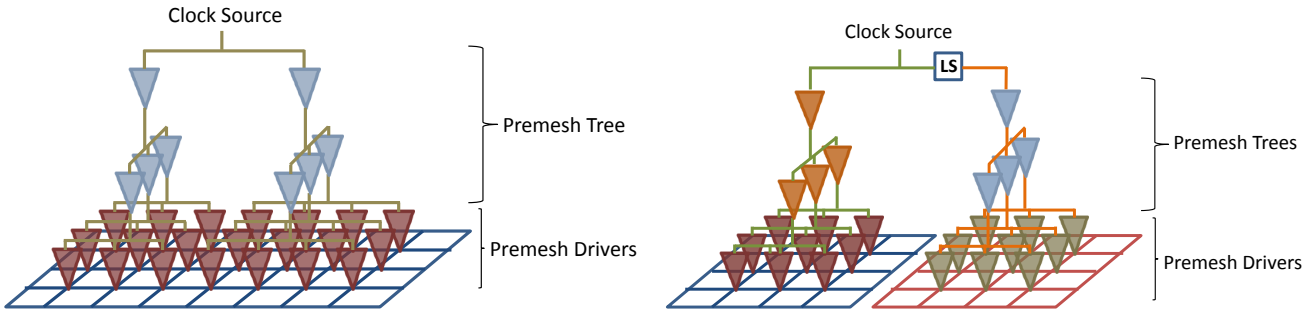
In Section II-A, clock mesh topologies and their challenges are described. In Section II-B, multi-voltage clocking techniques and their challenges are explained.

A. 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.

1) Power Dissipation in Clock Mesh: The power consumption of a clock mesh can be approximated as follows:

$$P_{total} \approx \alpha c_{total} f V_{dd}^2 \quad (1)$$



(a) Typical single-voltage domain clock mesh topology.

(b) Proposed multi-voltage domain clock mesh topology.

Fig. 1. Clock mesh topologies

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

$$c_{total} \approx c_{mesh} + c_{stub} + c_{pmd} + \sum_{\forall i} c_i \quad (2)$$

where the last term stands for the total capacitance of the sink registers, and c_{mesh}, c_{stub} and c_{pmd} are the total capacitance of the mesh wires, stub wires and the premesh drivers, 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 [4, 6, 7] propose mesh wire reduction, [5] proposes stub wire reduction and [8, 13] 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 (that are addressed in this paper), it is more promising than capacitance reduction because 1) most of the switching capacitance is at the clock sinks, and thus, mesh or stub wirelength 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 and not for the clock itself. It is an added incentive that these multiple voltage domains can help reduce the dissipation of a clock mesh structure, while simultaneously preserving the low skew property.

2) Skew in Clock Mesh: The global clock skew in the mesh is estimated as:

$$t_{skew} = t_{skew}^{pmd} + t_{skew}^{mesh} + t_{skew}^{stub} \quad (3)$$

where t_{skew}^{pmd}, t_{skew}^{mesh} and t_{skew}^{stub} are the skews introduced by the difference between the maximum and the minumum delays on the premesh drivers 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}^{pmd} is a challenge for multi-voltage meshes as the premesh trees in Figure 1(b) are disjoint.

B. Multi-Voltage Clocking Techniques

Multi-voltage designs necessitate the use of different clocking techniques due to the following challenges. First, the delay in the low voltage domains increases due to the increase in the gate delay of the buffers and the propagation delay of the wires. To that end, more delay must be added at the higher voltage domains to match the delay, and more repeaters must be added at the lower voltage domains to keep the signal integrity. Furthermore, level shifter cells are needed for cross domain clock delivery. All these design requirements introduce an extra power overhead which needs to be analyzed versus the power savings obtained through multi-voltage design.

III. Methodology

A new methodology to synthesize a mesh-based clock distribution network in a multi-voltage design flow is proposed. The steps of the proposed methodology are as follows:

1) The physical placement in each voltage domain is performed using standard EDA tools.
2) A uniform mesh size is selected for each voltage domain considering the skew constraints in the domain.
3) Premesh drivers are placed at each intersection, and sized considering the slew constraints.
4) Premesh tree is synthesized iteratively to decrease the global skew under a user-specified threshold.

In the clock mesh design, uniform clock mesh grids are preferred to place the mesh wires between power rails to prevent the crosstalk. Thus, uniform clock mesh grids are considered in this paper. In the typical single clock mesh design, shown in Figure 1(a), the mesh wires are directly connected to the sinks and the mesh is driven by a premesh tree. For the multi-voltage clock mesh, each domain needs its own clock mesh. To that end, local meshes are placed for each domain with their own premesh trees. Then, the roots of each domain are connected to a master root through level shifters, as shown in Figure 1(b). The mesh size selection, buffer driver selection and sizing and the premesh tree synthesis procedures are presented in Section III-A, Section III-B and Section III-C, respectively. Section III-A and Section III-B are simple algorithms that do

not define the novelty of this work but complement the design flow. Note that, any alternative, more (or less) sophisticated method for mesh size selection and mesh driver sizing can be seemlessly integrated, which demonstrates the practicality of the proposed method for automation purposes. Furthermore, the algorithms to be presented are designed for any number of voltage domains, however it is important to note that the number of voltage domains is typically 2 in practical applications.

A. Mesh Size Selection

In Eq. (3), t_{skew}^{mesh} and t_{skew}^{stub} are directly related to the density of the mesh. With a sparser mesh, the distance from the intersection (i.e. buffer locations) to the mesh connection of a stub wire and the distance from the sink register to the mesh connection point increases which contributes to t_{skew}^{mesh} and t_{skew}^{stub}, respectively. On the other hand, a denser mesh increases the mesh capacitance, which also increases the power consumption. To that end, the smallest N×N mesh size is targeted in each voltage domain by a simple exhaustive search by increasing N until the skew becomes less then a user defined threshold. If prior information is available on a suitable clock mesh size or depending on the floorplan size, a larger initial mesh size N can be selected to start the iterations. This procedure is summarized in Algorithm 1.

Algorithm 1 Mesh Size Selection

Input: Sink locations, skew constraint $skew_{const}^{i}$ for each voltage domain i
Output: The mesh size N_i for each voltage domain i
  for Each voltage domain i **do**
    Initialize N_i, e.g. N_i=1
    Calculate $skew_{max}^{i}$
    while $skew_{max}^{i} > skew_{const}^{i}$ **do**
      $N_i = N_i + 1$
      Update $skew_{max}^{i}$
    end while
  end for
  Return N_i for each domain i

B. Premesh Driver Selection

After the mesh size is selected, the premesh drivers are placed at each intersection. A simple heuristic is used to select these buffers. First, minimum buffers are placed at each intersection, and the buffer sizes are iteratively increased to meet the slew constraint. The slew constraint is typically set to 5% of the clock period. To that end, the user-defined slew constraint, $slew_{const}$, is selected as 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 up at the same time with an identical scale for the sake of simplicity. This procedure is shown in Algorithm 2.

Algorithm 2 Premesh Driver Selection

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

C. Premesh Tree Synthesis

Premesh tree synthesis is the most critical part of the multi-voltage clock mesh design flow, as the insertion delay at all voltage domains must be matched to preserve the zero skew operation. Moreover, as the slew of the mesh (Section III-B) is not completely independent from the premesh tree, the slew of the mesh should not be degraded when building the premesh tree.

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. In this work, an iterative algorithm is proposed to accomplish this task. In the algorithm, the premesh tree is built up iteratively without degrading the slew at the sink registers until the skew among the voltage domains becomes less than a user-specified threshold. This procedure is shown in Algorithm 3.

In Algorithm 3, first, the (n-1)st level of the premesh tree is synthesized considering the slew constraints (step 1). The purpose is not to degrade the slew set in Algorithm 2 more than 20% (note that overall slew limit is determined as a combination of the slew constraint $slew_{const}$ in Section III-B and this degration margin in Algorithm 3). Next, the insertion delay of each register sink is calculated to observe the delay differences among the voltage domains (step 2). The skew among the sink registers inside the same voltage domain can be set to zero (or almost zero) by setting the skew constraint, $skew_{const}^{i}$, in Algorithm 1 tight for each voltage domain i. Thus, the maximum insertion delay in each voltage domain is selected as the common insertion delay of that domain without loss of generality. In step 3, the iterations start in a while loop until the difference between the maximum insertion delay and the minimum insertion delay is less than the user-defined cross-domain skew threshold $skew_{const}^{cross}$. In step 4, the voltage domain k with the minimum delay is selected and a minimum buffer is added into its path iteratively (steps 6-9) until the delay K of this domain exceeds the minimum acceptable value. If K is larger than the maximum acceptable

Algorithm 3 Cross Domain Delay Matching

Input: Mesh size of each domain, assigned voltage for each domain, buffer library for each domain, slew constraint $slew_{const}$, cross domain skew constraint $skew_{const}^{cross}$

Output: Premesh tree for each domain i

```
Place the (n-1)st level of the premesh tree using slew_const
Calculate the maximum delay delay^i_max for each voltage island i
while max_∀i(delay^i_max) − min_∀i(delay^i_max) > skew^cross_const do
    K = min_∀i(delay^i_max) and k as the corresponding domain
    Set M = max_∀i(delay^i_max)
    while K < M − skew^cross_const/2 do
        Add a minimum buffer to k
        Update K = delay^k_max
    end while
    if K > M + skew^cross_const/2 then
        Remove the last-added buffer from k
        Update K = delay^k_max
    end if
    while K < M do
        Increase the size of the minimum buffer in domain k
        to the one larger buffer in the library
        Update K = delay^k_max
    end while
end while
Return the premesh tree for each domain i
```

value, the last-added buffer is removed from K, otherwise no change is necessary as K is already in the desired interval. In step 14, the delay of the voltage domain, K is compared with the maximum delay value M. If it is smaller than M, K is increased slowly by increasing the buffer sizes in voltage domain k starting from the buffer with the minimum size, until it exceeds M (steps 14-17). After this step, the same procedure follows on the voltage domain with the second minimum delay. The iterations stop when the global skew is less than the user-specified threshold.

In this procedure, the solution quality depends on the user-defined cross-domain skew threshold $skew_{const}^{cross}$, as it is the key parameter that controls the tradeoff between the skew and the power consumption (tight skew constraints lead to more inserted buffers). The global skew constraint is typically 2% of the clock period. To that end, the user defined global skew constraint $skew_{const}^{cross}$ is selected as the 2% of the clock period, although it can be set to a different value depending on the performance requirements.

IV. Case Study

The proposed methodology is implemented on a case study to verify the quality of the results. The experimental setup for this case study is explained in Section IV-A and the results are presented in Section IV-B.

A. Experimental Setup

The proposed methodology is implemented with Tcl in order to inter-operate with standard EDA tools and tested on a case study with two voltage domains. Similar to most standard cell libraries, the selected *SAED 90nm EDK Library* [14] has two voltage levels, a high voltage at 1.2V and a low voltage at 0.8V. Consequently, the number of voltage domains in this case study is 2. Within the two domains, two of the largest circuits of ISCAS'89 benchmarks, s35932 and s38417, are placed. The RTL level designs are synthesized using *Design Compiler* of *Synopsys* and the placement of the circuits are done using *IC Compiler* of *Synopsys*. The larger design s35932, with 1728 sinks, is supplied with 0.8V and the smaller s38417, with 1564 sinks, is supplied with 1.2V without loss of generality. The cell utilization is set to 0.45 of the total area for both domains. The skew and the power analysis are done 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 compare the quality of results with the standard single-voltage mesh and multi-voltage clock tree, the following procedure is performed: First, the same circuit is sythesized with a single mesh whose mesh size and premesh drivers are selected using the proposed methods given in Section III-A and Section III-B. For consistency, the premesh tree is synthesized using *IC Compiler*. Next, a dual-voltage clock tree synthesized and optimized by *IC Compiler* of *Synopsys*. The user-defined constraints are selected as follows: The skew inside a voltage domain $skew_{const}^{i}$ is selected as 2 ps for both domains to keep the intra-domain skew almost zero. 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, and the cross domain (global) skew $skew_{const}^{cross}$ is selected as 40 ps, as explained in Section III-B and Section III-C, respectively.

In the multi-voltage designs, level shifters are necessary on the cross communication paths in the logic. The power overhead of these level shifters within the logic power budget is usually taken into account at the design planning stage. Similarly, the effect of this overhead on the clock network should be investigated to include the multi-voltage clocking cost. To that end, it is necessary to investigate whether a multi-voltage domain clock mesh is still effective over a single-voltage domain clock mesh. To analyze this effect, three test cases are created benchmarking 50, 250 and 500 cross-domain communication paths. The number of cross-domain communication paths are selected considering the fact that each domain contains approximately 1,500 registers and 6000 total gates. As the number of input/output ports on the two selected circuits, s38417 and s35932, are not high enough to perform all of these tests, additional input/output ports are created on random internal nets to model the communication paths. Naturally, this setup changes the functionality of the circuits but it successfully serves the purpose of this work to demonstrate the impact of cross-domain communication paths among multi-voltage clock and logic domains on power consumption.

B. Experimental Results

The synthesized dual-voltage domain mesh, single-voltage domain mesh and the dual-voltage domain tree are shown in Figure 2, Figure 3 and Figure 4, respectively. The proposed method outputs a 10×10 mesh for both of the partitions and selects NBUFFX4 from the SAED 90nm library as the premesh driver types. The maximum slew at the register sinks in the proposed dual-voltage mesh design in Figure 2 is 108 ps, which is within the allowed margin of 120 ps. The simulations show that the proposed multi-voltage domain mesh can reduce the power dissipation by 37.14% compared to the single-voltage domain mesh in Figure 3, with a slight increase in the skew to from 1 ps to 10 ps. Compared to the dual-voltage clock tree in Figure 4 synthesized and optimized by *IC Compiler*, the skew of the dual-voltage clock mesh is down to 10 ps from 76 ps at the expense of a 20.92% increase in the power dissipation. The level of the 20.92% increase is an expected increase, as the increase in the power consumption from a clock tree topology to a clock mesh topology for a single voltage domain implementation is observed to be ≈35% on benchmark circuits using the same standard EDA tools. Also note that, this is a comparison between a design highly optimized with advanced EDA tools and a design created using simple mesh size selection and buffer sizing algorithms. More sophisticated methods for mesh size selection and buffer sizing can be used to improve this overhead in an industrial implementation.

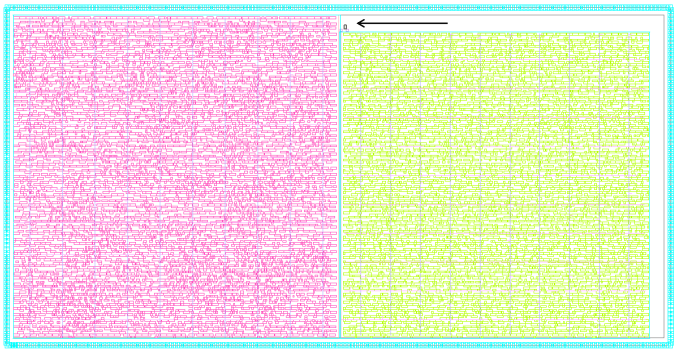


Fig. 2. Dual-voltage domain clock mesh; left partition operates at 1.2V and the right partition operates at 0.8V. 10×10 meshes are visible, and the placed level-shifter is highlighted at the top-middle.

At the time of this publication, standard EDA tools are not capable of automatically synthesizing a multi-voltage clock mesh. *IC Compiler 2011.09-SP2-1* can only optimize the premesh tree with a power objective. To emphasize the need for the automation presented in this work, the proposed multi-voltage clock mesh design is compared with the clock mesh design whose mesh size selection and the premesh driver sizing is done with the proposed method for consistency but whose premesh tree is synthesized in *IC Compiler* instead. As automated tools cannot match the delay at different voltage domains as necessitated by the proposed multi-voltage mesh implementation, the EDA tool optimizes the power with a larger delay mismatch. Proposed method can achieve substantially better skew at 10 ps (as low as 0.5% of the clock period) compared to 229 ps with an increase in the power dissipation by 8.5%. In order to highlight the power savings of the proposed method, it is also compared with a single voltage domain tree, synthesized and optimized by *IC Compiler*. The proposed method dissipates 18.27% less power compared to the single voltage domain clock tree with a skew of 78 ps. The power and the skew trade-off of the proposed dual-voltage mesh compared to the other implementations is summarized in Table I.

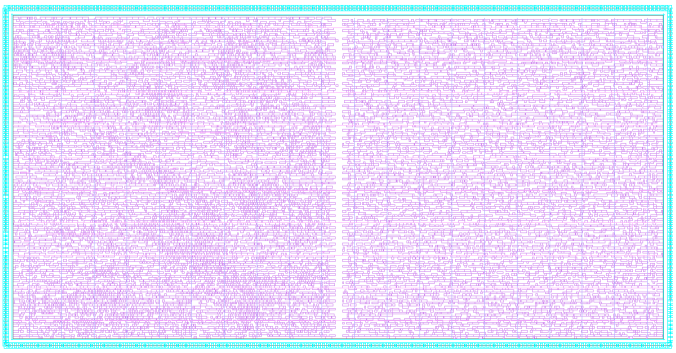


Fig. 3. Single-voltage domain clock mesh; both partitions operate at 1.2V. A 10×20 mesh is synthesized to cover both the regions.

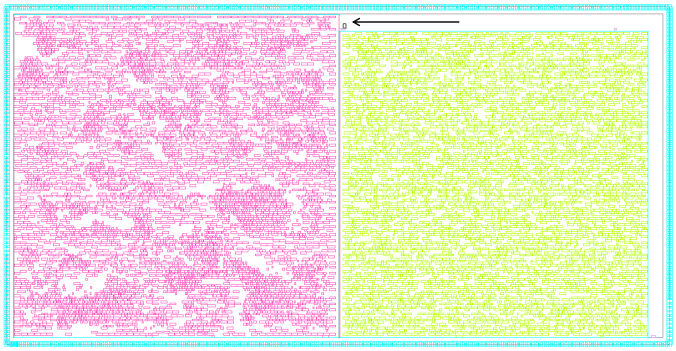


Fig. 4. Dual-voltage domain clock tree; left partition operates at 1.2V and the right partition operates at 0.8V. The placed level-shifter is highlighted at the top-middle.

In order to investigate the level-shifter overhead of the cross-domain logic paths on the clock network, the power consumption of the test cases with 0 (base case), 50, 200 and 500 paths are compared with the power consumption of the single-voltage domain clock mesh. The synthesized circuit with 50 cross-domain communication paths is shown in Figure 5. Note that a dual-voltage clock tree has the same communication cost for the same partitioning profile, therefore it is not included in this comparison. Moreover, the skew is the same for these four designs with different communication costs, as the clock network is identical for each design. The power dissipation increases with an increase in the number of paths as shown in Table II, which is expected. The results show that, multi-voltage clock mesh achieves a better power dissipation profile, even in the presence of high communication traffic among the domains. In particular, multi-voltage clock mesh can save 37.14% power for the base case,

TABLE I
THE POWER AND THE SKEW TRADE-OFF

Circuit	Skew (ps)	Power (mW)	Improvement compared to the existent methods	
			Skew (ps)	Power
Proposed dual-voltage mesh	10	67.23	-	-
Dual-voltage tree synthesized by ICC	76	55.60	66	-20.92%
Single-voltage mesh optimized by ICC	1	106.96	-9	37.14%
Dual-voltage mesh optimized by ICC	229	61.96	219	-8.50%
Single-voltage tree synthesized by ICC	78	79.51	68	15.44%

where there is no communication among the domains and 31.38% power for the case where there are 500 paths among the domains. This result shows that proposed multi-voltage domain clock mesh design has a wide application range from the SoCs that have few communication paths among the cores to the single core ASICs that are floorplanned with multiple voltage domains.

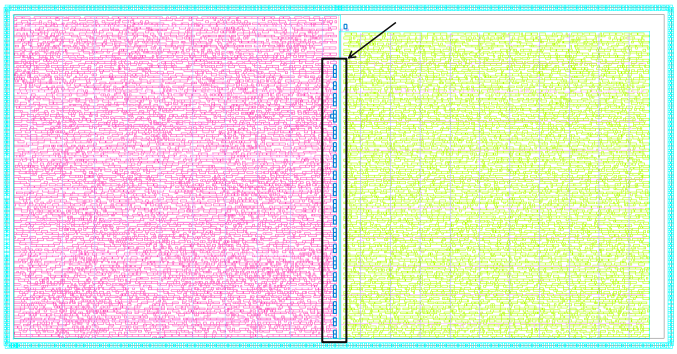


Fig. 5. Dual-voltage Domain Clock Mesh with 50 cross-domain communication path. The 50 level-shifters are highlighted in the middle.

TABLE II
THE EFFECT OF CROSS-DOMAIN (CD) COMMUNICATION PATHS

Circuit	Power (mW)	% Imp.
Single-voltage Mesh	106.96	-
Proposed MV Mesh with 0 CD Paths	67.23	37.14
Proposed MV Mesh with 50 CD Paths	67.40	36.99
Proposed MV Mesh with 250 CD Paths	70.00	34.55
Proposed MV Mesh with 500 CD Paths	73.39	31.38

V. CONCLUSIONS

A new methodology is proposed to synthesize multi-voltage, single clock domain clock meshes for high performance ICs. The proposed method enables clock mesh synthesis on designs with multi-voltage domains. The presented case study shows that multi-voltage mesh can achieve 37.14% better power compared to the single-voltage clock mesh with a slight increase in the skew to 10 ps (as low as 0.5% of the clock period) that remains within the skew budget. Even in the presence of high data communication among the voltage domains, proposed method dissipates 31.38% to 37.14% less power. Furthermore, it can achieve upto 66 ps better skew compared to the multi-voltage tree synthesized in advanced EDA tools with a limited increase in the power dissipation. In this work, simple algorithms are used for mesh size and premesh driver size selection to emphasize the elegance of multi-voltage clock mesh. Therefore, this methodology can easily be combined with existing clock mesh optimization and buffer sizing algorithms for improved results and a wide practical applicability. If more number of voltage levels is available, the proposed methodology can create more selections in the power vs. skew curve.

REFERENCES

[1] A. Rajaram, D. Z. Pan, and J. Hu, "Improved algorithms for link-based non-tree clock networks for skew variability reduction," in *Proceedings of the International Symposium on Physical Design (ISPD)*, 2005, pp. 55–62.

[2] 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.

[3] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing clock skew variability via cross links," in *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, June 2004, pp. 18–23.

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

[5] 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.

[6] A. Rajaram and D. Pan, "Meshworks: An efficient framework for planning, synthesis and optimization of clock mesh networks," in *Asia and South Pacific Design Automation Conference (ASPDAC)*, January 2008, pp. 250–257.

[7] 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.

[8] J. Lu, X. Mao, and B. Taskin, "Timing slack aware incremental register placement with non-uniform grid generation for clock mesh synthesis," in *Proceedings of the International Symposium on Physical design (ISPD)*, 2011, pp. 131–138.

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

[10] T. Xanthopoulos, D. Bailey, A. Gangwar, M. Gowan, A. Jain, and B. Prewitt, "The design and analysis of the clock distribution network for a 1.2 ghz alpha microprocessor," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2001, pp. 402–403.

[11] C.-C. Tsai, C.-C. Kuo, and T.-Y. Lee, "Voltage-island aware x-clock tree construction for power minimization," in *International Conference on Computer Science and Service System (CSSS)*, June 2011, pp. 4132–4135.

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

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

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