

Design and Optimization of Multiple-Mesh Clock Network

Jinwook Jung^(✉), Dongsoo Lee, and Youngsoo Shin

Korea Advanced Institute of Science and Technology, 291 Daehak-ro,
Yuseong-gu, Daejeon 305-701, Korea
jinwookjung@kaist.ac.kr

Abstract. A clock mesh, in which clock signals are shorted at mesh grid, is less susceptible to on-chip process variation, and so it has widely been studied recently for a clock network of smaller skew. A practical design may require more than one mesh primarily because of hierarchical clock gating architecture; a single mesh, however, can also support the same architecture after some hierarchies are removed but at the cost of gating efficiency. We experimentally compare multiple- and single-mesh using a few test circuits, and show that the former consumes smaller clock power (16.3%) but exhibits larger clock skew (10.2 ps) and longer clock wirelength (21.7%). We continue to study how multiple meshes should be floorplanned on the layout, specifically whether or not we allow the overlaps among meshes. The choice is translated into different physical design strategy, and causes different amount of clock skew, critical path delay, clock wirelength, and clock power consumption, which we experimentally evaluate. We give at last the comparison of clock skew variation for each mesh implementation and clock tree, and show that floorplanning of multiple meshes helps to reduce the variation of clock skew.

Keywords: Clock distribution · Clock mesh · Multiple-mesh clock network

1 Introduction

Big industrial designs such as SoCs and processors are often embedded with multiple levels of clock gating to efficiently reduce the power consumption of clock distribution network [1–3]. Some clock gating is inserted by automatic CAD tools, e.g. by compiling load-enable registers into normal registers driven by clock gating cells (CGCs); designers may also insert clock gating in manual fashion, especially at module- or system-level, based on the knowledge of the usage scenario of a design [4].

If the clock network of such a design is to be constructed using clock meshes to achieve lower clock skew, multiple meshes may be inserted as shown in Fig. 1. This is a natural choice in terms of power consumption because each mesh can be gated whenever the block it spans is not actively switching. Furthermore, it is

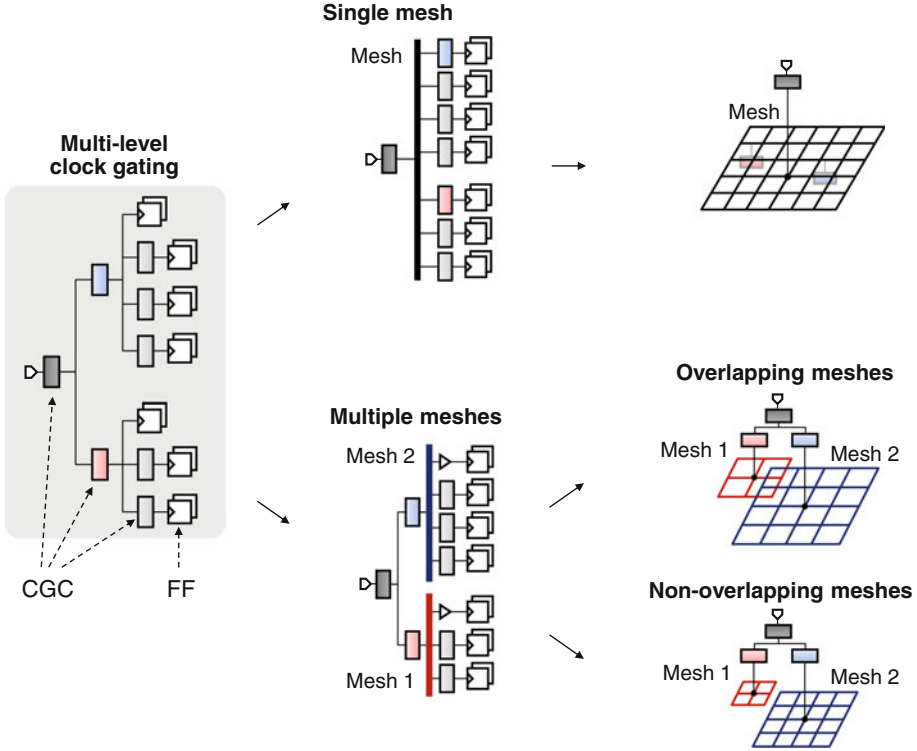


Fig. 1. Design of mesh clock network for multi-level clock gating.

well known that mesh consumes more power than standard clock tree network [5] due to more wire capacitance and excessive short-circuit current; a study indicates that 33.4% more power is consumed in comparison with the standard clock tree [6], so it helps to gate mesh whenever it is possible. A single big mesh, however, may be inserted instead after some clock gating hierarchies are removed, which is also illustrated in Fig. 1. This choice is not efficient in terms of power consumption, but it has the benefits of shorter design time because of its simpler structure, as well as shorter clock wires and more importantly smaller clock skew. In this paper, we quantitatively explore the two styles of mesh implementation, using some test circuits in 28-nm technology, which is the first contribution.

When multiple meshes are employed, it is important to decide how to floor-plan them. If overlaps between meshes are allowed, physical design can be done in flat. No overlap, on the other hand, implies hierarchical physical design. The two styles will have different impact on clock power, clock wirelength, clock skew, and timing closure, which we want to quantitatively assess; this constitutes the second contribution of the paper.

The remainder of this paper is organized as follows. The basic mesh network structure and the steps to synthesize it are reviewed in Sect. 2; clock gating

in multiple levels of hierarchy is also described. In Sect. 3, we address the procedures to design single- and multiple-mesh clock networks in the context of multi-level clock gating, and use some test circuits to experimentally assess the two implementation styles. Section 5 discusses the floorplan of multiple meshes and provides experimental evaluation. Section 6 gives the comparison of the three mesh implementations with the standard clock tree, and evaluates clock skew variation. Several related works are reviewed in Sect. 7, and we finally conclude the paper in Sect. 8.

2 Preliminaries

2.1 Clock Mesh Structure and Its Synthesis

Figure 2 illustrates a structure of mesh clock network that we are concerned with in this paper. It consists of three main components: a premesh tree, a mesh grid, and a postmesh tree. Clock sinks are connected to the mesh through postmesh buffers and stub wires. A premesh tree can be a balanced H-tree or standard clock tree, and connects the mesh to the clock source. Leaf-stage buffers in the premesh tree will be called mesh drivers.

Figure 3 illustrates the overall synthesis flow of a mesh clock network; we synthesize a mesh clock network in a bottom-up manner. Clock sinks are first grouped together based on their locations; maximum fanout of postmesh buffers, which are inserted and sized properly once the groups are formed, determines the group size. A mesh grid is constructed and connected to the postmesh buffers through stub wires; a grid structure is designated by the numbers of vertical and horizontal wires. It is determined in a way that minimum length of wires are used for mesh grid and stub wires. The pitch of grid wires that gives the

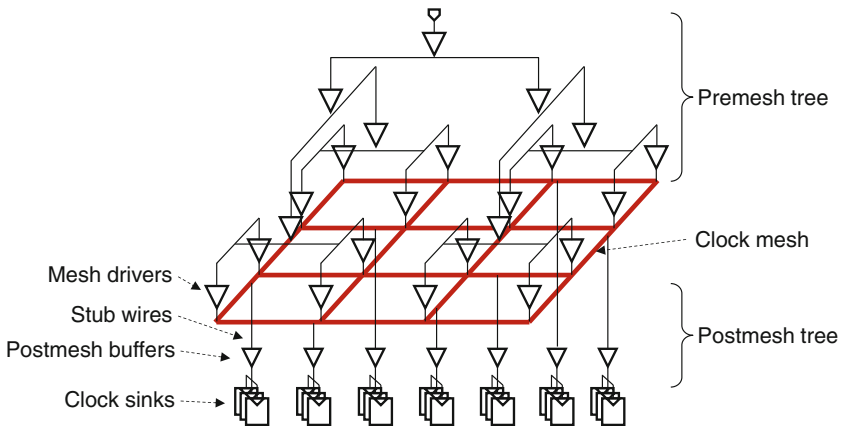


Fig. 2. Mesh clock network.

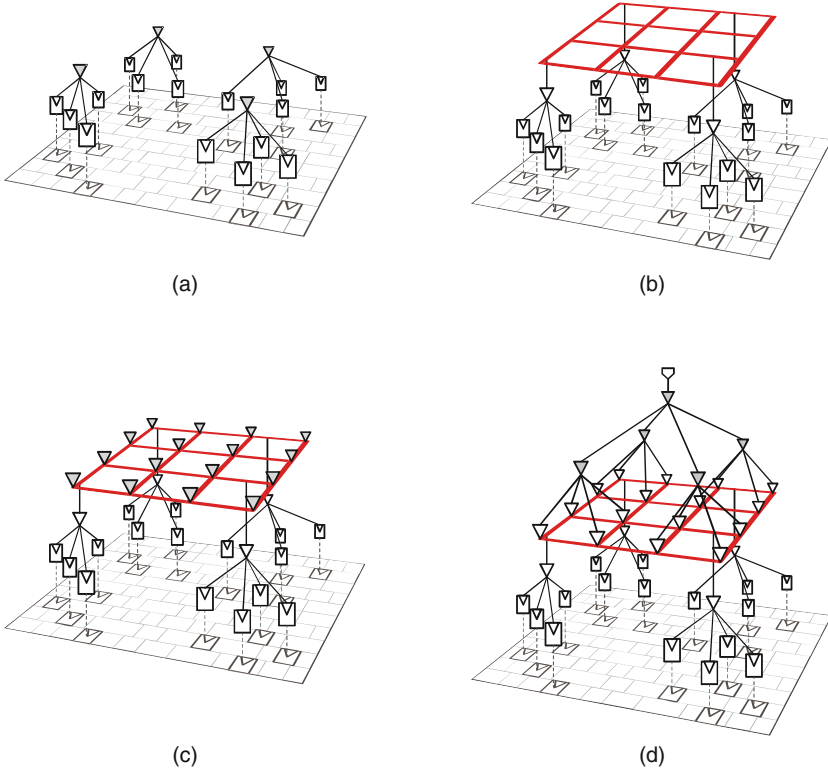


Fig. 3. Mesh clock network synthesis: (a) postmesh buffer insertion, (b) mesh grid construction, (c) mesh driver insertion, and (d) premesh tree synthesis.

mesh of the minimum wire length is calculated as [7]:

$$p = \sqrt{\frac{12}{\rho}}, \quad (1)$$

where ρ denotes the density of the postmesh buffers in the placement area. Then we can determine the number of vertical wires m and horizontal wires n :

$$m = \left\lceil \frac{W}{p} \right\rceil, \quad n = \left\lceil \frac{H}{p} \right\rceil, \quad (2)$$

where W and H are the width and height of the placement area, respectively. After the mesh grid construction, mesh drivers are placed at each grid location; they then serve as the sinks of premesh tree synthesis.

2.2 Multi-level Clock Gating

Clock gating is a standard technique to reduce clock power. It is often applied in multiple levels, particularly in big industrial designs [1–4]. This is illustrated

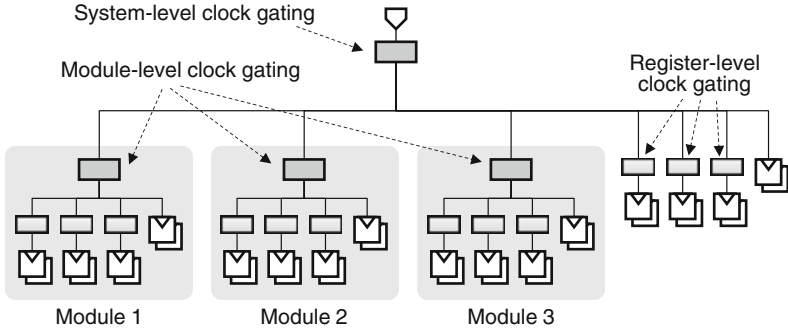


Fig. 4. Clock gating in multiple levels.

in Fig. 4. Register-level clock gating is mostly realized through automatic CAD tools, e.g. by replacing load-enable registers with clock gating cells (CGCs) and normal registers, and by employing XOR self-gating [8].

In addition, designers may explicitly instantiate CGCs at module level or system level (right after the clock source) according to the usage scenario of a chip. This type of clock gating gives the capability to turn off the clock signal of specific modules or entire systems, and shuts down a large portion of clock distribution network.

3 Mesh Clock Networks for Multi-level Clock Gating

A design of multiple level clock gating encounters the choice of mesh implementation styles as shown in Fig. 4. Specifically, a single big mesh may be inserted at system level (or at each clock domain) or more than one small meshes may be inserted with each mesh assigned to a module or to a group of registers. The two styles incur different clock power consumption, as well as different clock skew, wirelength, and design time, which we want to explore in this section.

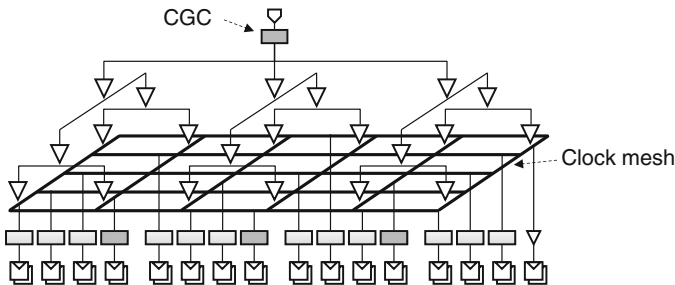


Fig. 5. Single mesh implementation of multi-level clock gating.

3.1 Single-Mesh Implementation

In this implementation, a single big mesh is inserted right after the system level clock gating of Fig. 4. The resulting clock network is shown in Fig. 5. To retain the advantage of smaller clock skew of mesh network, it is desirable to have short clock paths from mesh to each clock sink. But, multiple levels of clock gating after mesh (see Fig. 4) lend themselves to local clock trees with a few CGCs and buffers. The key therefore is to remove the hierarchy of clock gating so that the paths from mesh to clock sinks become shorter. The module-level CGCs are removed for this purpose; a new CGC is inserted to each group of registers that have directly been gated by a module-level CGC; a CGC that has been driven by module-level CGC is now gated by its original gating logic and the logic that has gated module-level CGC.

It is well known that mesh consumes more power than clock tree due to more wire capacitance and short-circuit current [5, 7]. It is thus important to gate mesh as often as possible. A single big mesh, however, is gated less frequently, thus has disadvantage in power consumption. Balancing postmesh trees should be easier, which yields smaller skew. Test circuits will be used to assess these factors, as well as wirelength and design time.

The maximum fanout of newly inserted CGCs, which serve as postmesh buffers, may be increased to reduce the additional capacitance of mesh wires; from Eqs. 1 and 2, reducing the number of postmesh buffers results in fewer mesh grid segments. Therefore we can consider using a small number of postmesh buffers to cut down the power consumption of clock meshes. However, it leads us to choose postmesh trees containing multiple levels of buffers which incur additional clock skew, since the maximum transition constraint of clock signal may be violated without a buffered tree structure. We will see how the maximum fanout of newly inserted CGCs affects clock skew and power consumption of the single-mesh implementation later in this section.

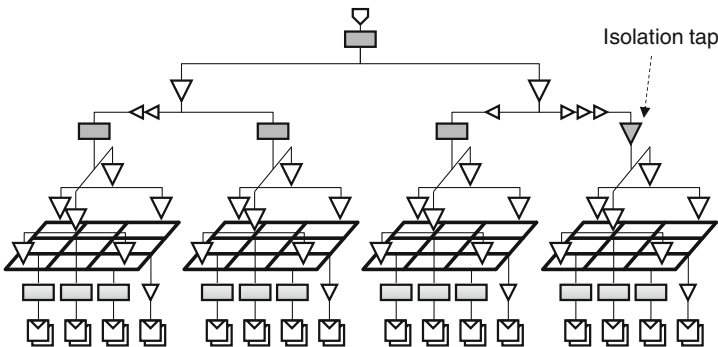


Fig. 6. Multiple mesh implementation of multi-level clock gating.

3.2 Multiple-Mesh Implementation

Another implementation of mesh is shown in Fig. 6. This time, a mesh is assigned to each module as well as to registers that have not belonged to any modules, which we call top-level registers. The initial clock network shown in Fig. 4 may be very unbalanced; in particular, the path from the clock source to top-level registers tends to be shorter. This is alleviated by inserting isolation taps, which have comparable delays to CGCs. If there are some modules without module-level clock gating, their clock sinks are also isolated by the isolation taps. Mesh drivers are inserted at each grid of meshes; they are then considered as sinks of premesh tree synthesis.

Since each mesh is gated at module-level, it can be gated more frequently, and leads to smaller power consumption. Clock skew can arise between different meshes as well as between different clock sinks under the same mesh; so skew is very likely to be larger than that in a single mesh implementation. Design complexity and wires will also increase.

3.3 Assessment

The design flow of mesh network synthesis for single- and multiple-mesh implementation has been implemented in Tcl, which runs on commercial placement and route tool; it is illustrated in Fig. 7. To determine the number of mesh wires in each implementation, we used Eq. 2. A few test circuits have been chosen from OpenCores [9]; the RTL description of each circuit has been modified to insert module- and system-level clock gating. A library of 28 nm industrial technology has been used to compile each circuit and to obtain a netlist. The last column of Table 1 corresponds to the number of meshes when clock is implemented as multiple meshes; the numbers of gates and flip-flops are also shown. Clock skew and power consumption have been measured using SPICE after parasitics are extracted from layout.

Table 1. Test circuits

Circuits	# Gates	# FFs	# Meshes
ac97	3225	1067	4
mc	6211	1069	3
usbf	7647	1736	3
pci	11142	3206	4
sdc	11815	3760	5
spi	13964	4656	2
des3	63217	8811	4
fft64	71263	15996	4

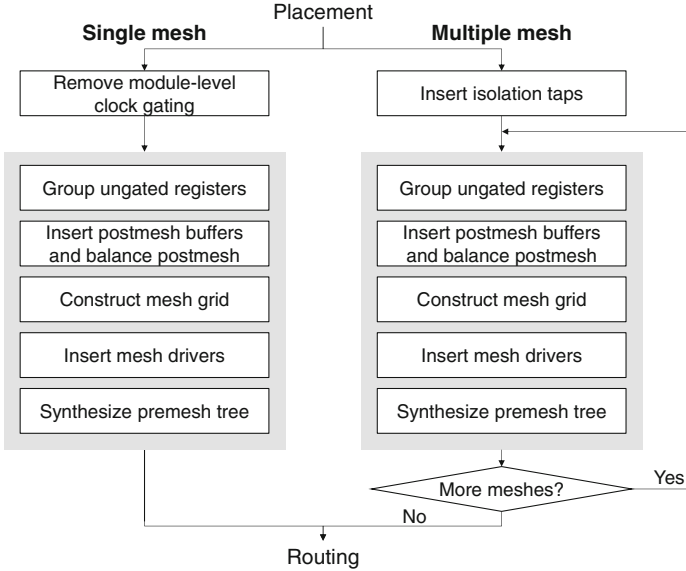


Fig. 7. Design flow of mesh network synthesis.

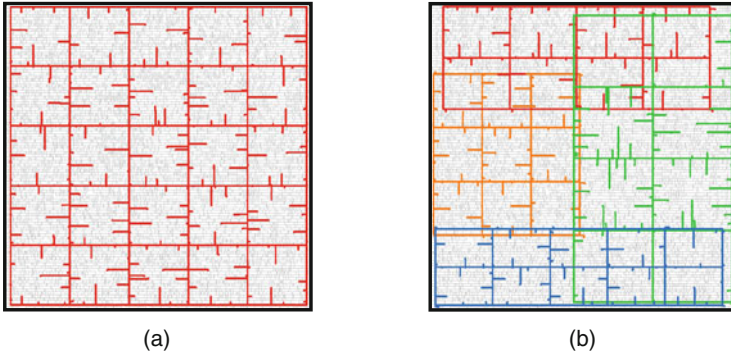
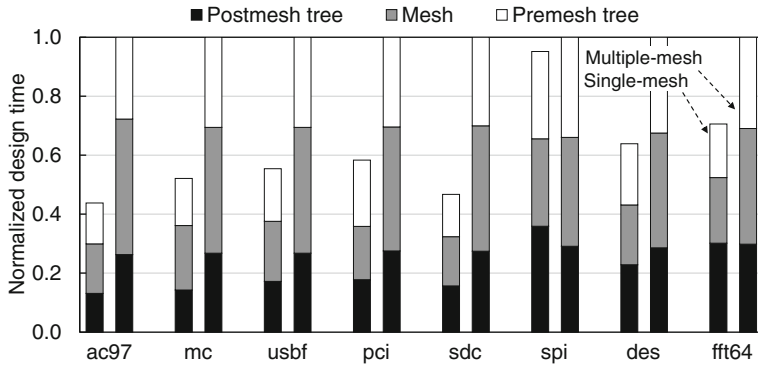
Comparison of Single- and Multiple-mesh Implementations. Single- and multiple-mesh implementations are compared in Table 2. Multiple meshes consume on average of 16.3% smaller power than single mesh. This has been expected because small multiple meshes are gated more often than a single big mesh; meshes are gated 78% of time in multiple meshes (on average of meshes, and on average of circuits), while a single mesh is gated 49% of time. Relatively small difference in power, considering the big difference in mesh gating probability, is due to more clock wires in multiple-mesh implementation as indicated in columns 5–7. Figure 8 depicts how respective mesh grids of single- and multiple-mesh networks are constructed in circuit ac97. Multiple meshes are placed as overlapped each other due to irregular module boundaries, causing the sum of mesh wires to be increased by 21.7% compared to single mesh on average of the circuits.

Clock skew is compared in the last three columns. It clearly shows the advantage of the single-mesh implementation of which clock skew is 10.2 ps smaller than that of the multiple meshes on average, which also has been expected. Postmesh buffers are close to a mesh grid in the single-mesh while the stub wires of the multiple-mesh implementations becomes longer (see Fig. 8(b)), which introduce additional clock skew due to the stub wire delay. Also, different meshes themselves contribute to clock skew in the multiple-mesh implementation due to the different latency from the clock source to each mesh (see Fig. 6).

Figure 9 compares the time elapsed for clock network synthesis. Multiple-mesh implementation takes 35.4% more time than single-mesh, on average. This is mainly due to the fact that designing mesh grid and postmesh trees has to be

Table 2. Comparison of single- and multiple-mesh implementation

Circuits	Clock power (mW)			Clock wirelength (mm)			Clock skew (ps)		
	Single	Multiple	Diff. (%)	Single	Multiple	Diff. (%)	Single	Multiple	Diff. (ps)
ac97	0.52	0.43	17.1	5.5	6.6	-20.0	13.5	26.6	-13.1
mc	0.18	0.15	18.2	4.8	6.3	-31.7	13.4	20.9	-7.5
usbf	0.89	0.85	10.0	7.6	10.0	-31.8	11.8	27.0	-15.2
pci	0.50	0.45	11.8	13.9	17.8	-28.7	13.4	26.1	-12.7
sdc	0.45	0.32	28.2	14.3	17.6	-23.6	14.0	23.3	-9.3
spi	0.84	0.62	26.1	18.3	20.2	-10.8	12.5	19.6	-7.0
des3	2.95	2.55	13.6	38.2	45.5	-19.4	14.0	24.5	-10.5
fft64	1.74	1.55	10.9	62.6	67.7	-8.0	19.8	26.0	-6.2
Average			16.3			-21.7			-10.2

**Fig. 8.** Layouts of (a) single- and (b) multiple-mesh implementations for circuit ac97.**Fig. 9.** Comparison of design time.

iterated in the multiple-mesh implementation. A circuit spi is an exception. It contains only two meshes in its multiple-mesh implementation; more times are spent in the postmesh tree synthesis of single-mesh implementation due to the large number of clock sinks (in consideration of circuit size).

Impact of Using Fewer Postmesh Buffers. We took the circuit ac97 and implemented two more single-mesh clock networks with two and four times bigger maximum fanouts of newly inserted CGCs, respectively, to see how reducing postmesh buffers affects clock skew and power consumption of the single-mesh implementation. The respective postmesh trees are now synthesized as 2-level and 3-level clock trees. The measured clock power, wirelength, and clock skew are summarized in Table 3 along with the results of single- and multiple-mesh implementations of the previous section.

Table 3. Experimental results for various postmesh trees of ac97

Scheme	Clock power (mW)	Clock wirelength (mm)	Clock skew (ps)
Single	0.52	5.5	13.5
w/ 2-level postmesh	0.47	5.3	33.1
w/ 3-level postmesh	0.43	5.0	51.3
Multiple	0.43	6.6	30.6

Figure 10 shows the clock mesh layouts. The number of grid segments are reduced due to the increased fanout of mesh grid; the clock wirelengths of the single-mesh network with 2- and 3-level postmesh trees are decreased by 3.3% and 8.7% compared to the original single-mesh implementation. It results in lower power consumption; the power consumption of the single-mesh implementation with 3-level postmesh is now close to the multiple-mesh implementation

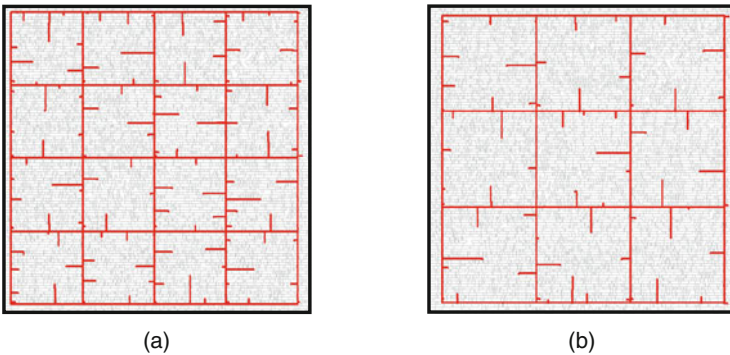


Fig. 10. Layouts of single-mesh networks of different postmesh structures.

(see column 2 of Table 3). As the depth of postmesh trees becomes deeper, however, clock skew is increased; it is now even larger than the multiple-mesh implementation, and the benefit of using single-mesh implementation in terms of clock skew diminishes. Therefore, it is better to choose the multiple-mesh implementation over the single-mesh network with deeper-levels of postmesh trees for lower power consumption.

4 Choosing Mesh Implementation Style

Assessments in Sect. 3.3 indicate that a single big mesh has advantages over a multiple-mesh network in terms of clock skew. On the other hand, the multiple-mesh implementation shows reduced power consumption due to the capability of shutting down a large portion of clock network; a low-power design may take multiple-mesh as the design strategy of choice.

Although our evaluation results show that all the test circuits taken for the assessments consume lower power in the multiple-mesh implementation, they also result in longer clock wirelengths. This fact implies the excessive metal resources may cause power overhead when gating probabilities of CGCs in a design are small. Here, let us briefly address the impact of gating probabilities in power consumptions. We took the circuit ac97 and generated SPICE netlist of single- and multiple mesh clock networks with parasitics extracted. To see how gating probabilities affect the power consumption, we controlled the enable signals arbitrarily and estimated power consumptions of different gating scenarios.

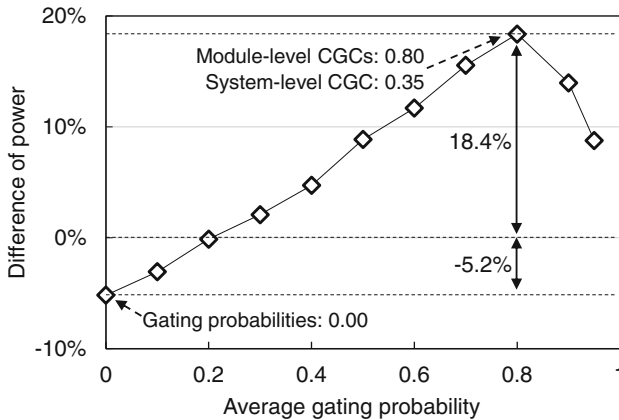


Fig. 11. Difference in power consumption between two mesh implementations for ac97 with respect to gating probabilities.

Figure 11 plots the difference of power consumption between two design options with respect to different gating probabilities; the difference is calculated by subtracting the power of single-mesh from that of multiple-mesh. If a circuit

does not gate at all, a single big mesh consumes lower power due to shorter wirelength. As the gating probabilities become larger, multiple mesh implementation begins to have smaller power consumption. The difference of power consumption has the maximum at average gating probability of 0.8. As the gating probabilities are still more increased, the power advantage of multiple-mesh implementation begins to shrink; this is because system-level clock gating also has large gating probability in that case.

4.1 Switching Capacitance Estimation

The mesh implementation of choice depends on the gating probabilities in a design. It may raise the question of how we know which mesh network has a benefit of power consumption. If there is a method of estimating switching capacitance of two strategies, we can select the mesh network of lower power before mesh construction; power is proportional to switching capacitance as is well known.

Let ΔC be the difference of the total capacitances in single- and multiple-mesh implementations. The following equation then allows us to select the suitable design strategy of mesh network before actual mesh construction:

$$\Delta C = k \left(\alpha_s C_m - \sum_{\forall m_i} \alpha_i C_m^i \right), \quad (3)$$

where k is an empirical constant, α_s and α_i are the switching activities of system- and module-level clock gatings, C_m is the capacitance of a single big mesh, m_i is the i th mesh in multiple-mesh, and C_m^i denotes the capacitance of the i th mesh of multiple-mesh implementation.

The structure of a mesh and the area it spans can be known just after the placement stage. So we can calculate the capacitance involved in a mesh from Eq. 2 and capacitance per unit length. Eq. 3 expresses the difference in switching capacitance of mesh clock network between the single- and multiple-mesh implementations. To consider premesh tree capacitance, we multiply an empirical constant k (e.g., 1.75 for H-tree); the wirelength of premesh tree is almost proportional to the size of mesh grid, as shown in Fig. 12. Functional simulation at earlier design stage provides the gating probability. If α_i s are relatively small, ΔC can be negative; the high gating probabilities will yield the positive value of ΔC . Therefore, we can evaluate the equation Eq. 3 and predict which implementation will have smaller power consumption without actually constructing the two mesh clock networks.

We will not take up this matter further in this paper since our assessments in Sect. 3.3 show that gating probability is relatively high, and multiple-mesh is always better in terms of power consumption for the test circuits. Nevertheless, if the functional simulation at earlier design phase indicates that the design has smaller value of gating probability, designers may consider the adoption of single-mesh for lower power.

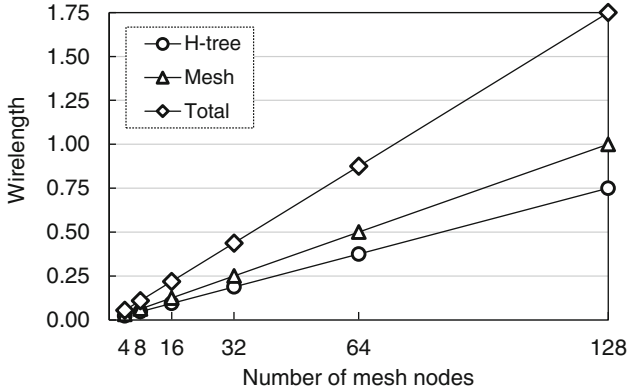


Fig. 12. Estimation of k value for H-tree premesh.

5 Floorplanning of Multiple Meshes

It has been shown in Sect. 3 that multiple mesh implementation has advantage in clock power even though it incurs longer clock wirelength and larger clock skew. In this section, we want to explore how multiple meshes can be floorplanned. Specifically, we may or may not allow the overlaps between meshes¹ as shown in Fig. 13. Note that the overlap does not cause the use of additional metal wires as illustrated in Fig. 14.

The choice of mesh floorplanning has significant implication in physical design process. Figure 13(a) allows flat placement and routing, thus more flexibility in achieving timing closure even though more wires will be used for mesh grid; Fig. 13(b) on the other hand assumes hierarchical physical design which is associated with more design steps and less design flexibility, but with less usage of wires for mesh grid. We want to experimentally assess the two choices in terms of clock power, clock wirelength, clock skew, and critical path delay.

5.1 Assessment

When overlap is allowed, placement is performed in flat. The region is identified from the location of flip-flops that belong to the same mesh, and mesh grid is constructed accordingly. The remaining steps of mesh network synthesis follow those of Sect. 3.2. For meshes without overlap, floorplanning is performed manually by referring to the relative locations of meshes with overlap (i.e. obtain Fig. 13(b) from 13(a)). We then assign a bounding box to all flip-flops and combinational gates that belong to the same mesh. Automatic placement is then performed with a set of bounding boxes as placement constraints, which is followed by mesh network synthesis.

¹ We wanted to compare single- and multiple-mesh implementation using the same placement, so overlap was allowed in Sect. 3.

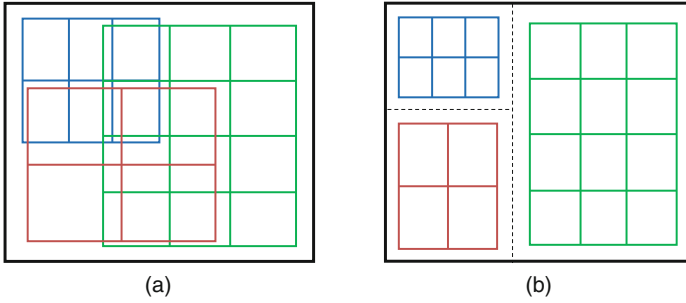


Fig. 13. Floorplanning of multiple meshes: (a) with overlap and (b) without overlap.

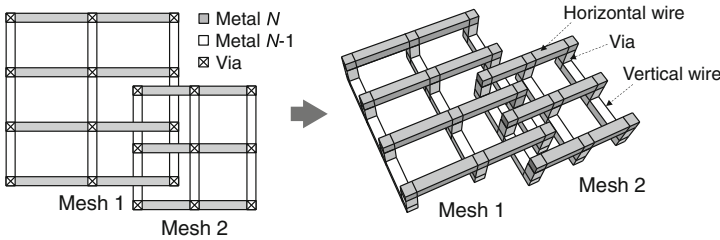


Fig. 14. Three-dimensional illustration of overlapped clock meshes.

The two mesh floorplanning methods are compared in Table 4. Floorplanning without overlap yields smaller clock power (5.2% on average), which is mainly due to shorter clock wirelength (15.9% on average). A circuit `usbf` is an exception, i.e. clock power is not very different even with large difference in clock wirelength. Its meshes are not gated very often (28% of time); it consists of one big mesh and two small meshes, so large number of buffers are inserted to balance clock arrival time to three meshes, much more when overlap is not allowed.

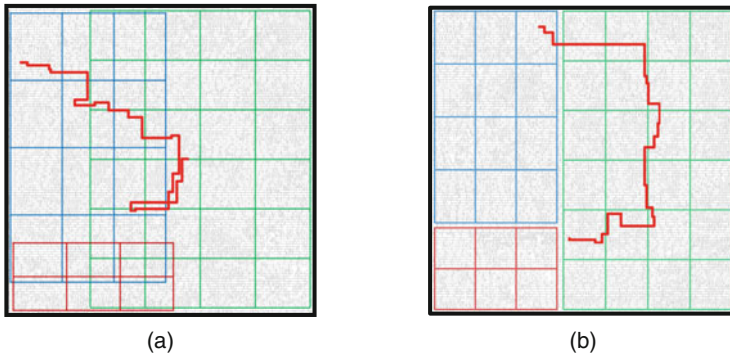
Table 4. Comparison of overlapping and non-overlapping meshes

Circuits	Clock power (mW)			Clock wirelength (mm)			Clock skew (ps)		
	Overlap	No overlap	Diff (%)	Overlap	No overlap	Diff. (%)	Overlap	No overlap	Diff. (ps)
<code>ac97</code>	0.43	0.43	1.1	6.6	5.8	11.1	26.6	18.3	8.3
<code>mc</code>	0.15	0.13	12.0	6.3	4.8	24.0	20.9	20.1	0.7
<code>usbf</code>	0.85	0.83	2.4	10.0	7.6	23.4	27.0	25.5	1.5
<code>pci</code>	0.45	0.39	12.8	17.8	14.0	21.4	26.1	20.0	6.1
<code>sdc</code>	0.32	0.31	3.5	17.6	14.9	15.8	23.3	20.6	2.7
<code>spi</code>	0.62	0.60	4.1	20.2	18.1	10.5	19.6	13.4	6.2
<code>des3</code>	2.55	2.50	1.8	45.5	38.1	16.2	24.5	16.3	8.2
<code>fft64</code>	1.55	1.48	4.2	67.7	64.1	5.3	26.0	21.6	4.4
Average			5.2			15.9			4.8

Table 5. Critical path delays of multiple mesh designs

Circuits	Overlap (ns)	No overlap (ns)	Diff. (ns)
ac97	1.70	1.72	-0.01
mc	3.18	3.21	-0.03
usbf	2.12	2.30	-0.18
pci	2.59	2.70	-0.11
sdc	2.67	2.77	-0.10
spi	2.70	2.82	-0.11
des3	2.43	2.46	-0.03
fft64	3.84	4.07	-0.23
Average			-0.10

Clock skew becomes smaller when overlap is not allowed; it is reduced by 4.8ps average. Meshes are smaller in this case (see Fig. 13), so mesh grid pitch also becomes smaller; the longest stub wire, which affects the skew, becomes shorter as a result.

**Fig. 15.** Critical paths in usbf: (a) meshes with overlap and (b) meshes without overlap.

We have also measured the critical path delay, which are reported in Table 5. It is clearly shorter when overlap is allowed (0.10 ns), because placement is performed in flat with greater flexibility in meeting circuit timing. Figure 15 illustrates how critical paths are identified in two mesh floorplans of the circuit usbf.

6 Comparison with Clock Tree

In this section, we compare the three mesh implementation styles, that we have covered, with the standard clock tree. We implemented a clock tree in each test

Table 6. Experimental results of clock trees

Circuits	Clock power (mW)	Clock wirelength (mm)	Clock skew (ps)
ac97	0.37	4.5	32.2
mc	0.13	4.2	39.9
usbf	0.74	7.1	52.6
pci	0.34	12.4	53.6
sdc	0.31	12.2	58.3
spi	0.57	15.6	54.4
des	2.33	31.9	69.6
fft64	1.22	54.8	69.2

circuit for this purpose using the commercial placement and route tool. Clock power, wirelength, and clock skew of the clock trees are reported in Table 6.

Figure 16(a) shows the clock skew of each clock network (normalized to the clock of the clock tree). Compared to clock tree, a 39.7 ps reduction of clock skew is achieved by adopting the single-mesh implementation. Two multiple meshes also significantly improve clock skew; 29.5 ps and 34.3 ps reductions are observed in multiple meshes with and without overlap, respectively. Note that the benefit of reducing clock skew by clock mesh grows as the number of clock sinks becomes larger; the divergence of clock paths increases, so the clock skew of a clock tree tends to increase. On the other hand, a large number of clock sinks share the clock path in a mesh clock network. Delay balancing between different meshes should be done for different meshes in the multiple-mesh implementation, but it is easier than in the clock tree since there are a few clock path to be balanced.

Power consumptions are also compared in Fig. 16(b) (normalized to the single-mesh implementation). As is well known, clock trees shows less power consumption than mesh networks; it needs 29% less power than a single mesh on average. Multiple meshes can reduce this large power overhead of clock mesh; floorplanned multiple meshes consume only 8% larger power than clock tree.

6.1 Clock Skew Variation

We generated the SPICE netlists of the three mesh implementations and a clock tree from the circuit ac97 with parasitics extracted, and conducted Monte Carlo simulation of 1,000 samples to evaluate clock skew variation. We obtained the arrival times of all clock sinks, and calculated the global clock skew by subtracting the minimum arrival time from the maximum arrival time.

Figure 17 shows the histogram of the clock skew for each clock network. Single-mesh network and clock tree show the smallest and the largest clock skew variations, respectively. Multiple-mesh implementation with mesh overlap is affected on-chip variation more than multiple meshes without overlap. It is due to its longer latency; a mesh network of overlapped multiple meshes has

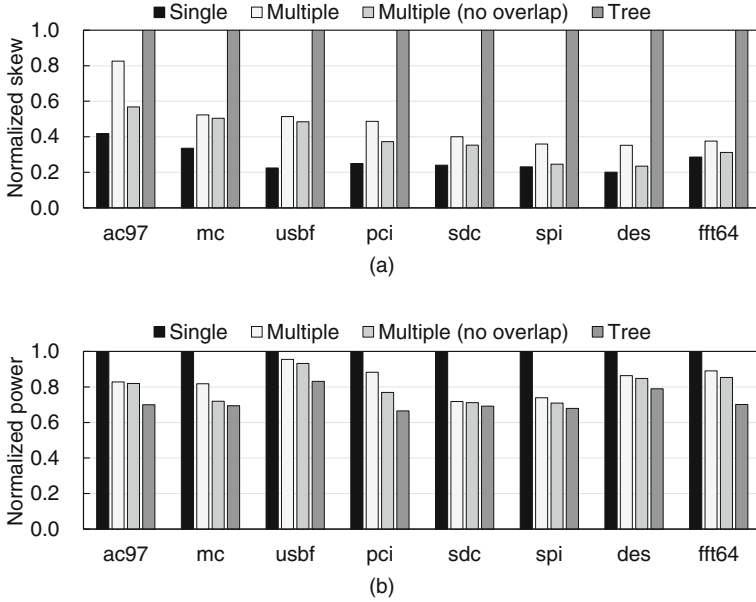


Fig. 16. (a) Normalized clock skew and (b) clock power.

longer wirelength and thereby more wire capacitance, so it shows larger clock latency than floorplanned meshes. Floorplanning the multiple meshes reduces the clock wirelength, so it can reduce the clock skew variation.

7 Related Work

There have been various studies concerning mesh clock network, particularly on the reduction of its excessive power consumption. A representative method is to reduce the wire usage of a mesh clock network thereby the wire capacitance. Such approach can be divided into two big categories; one is the reduction of unnecessary mesh grid segments [10], and another approach is the shortening the stub wires by moving clock sinks or grid wires [11, 12]. Short circuit current is also an important source of power consumption in mesh clock network, so several researches have proposed dedicated mesh driver to cut off the short circuit current [7].

However, there are few studies on mesh network design considering the clock gating although it is a pervasive technique to reduce clocking power. Lu et al. proposed a mesh clock network with several gated local trees [13]. They grouped FFs in the same grid after the mesh grid construction, and extracted gating function from the FF group. However, there is the limit to extract gating functions from only the adjacent FFs in the same grid box. Also, their methodology is impractical since in most cases the clock gating structure is defined before the

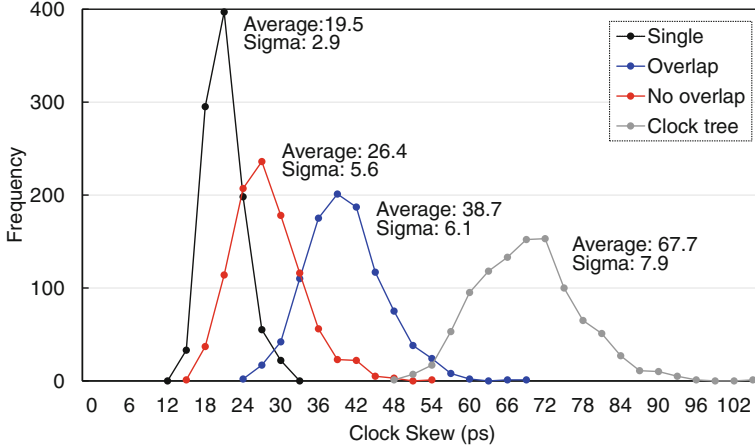


Fig. 17. Clock skew histogram.

placement stage. Wilke and Reis [14] compared clock skew and power consumption of a multiple-mesh network with a single-mesh network. They concluded that although the former has greater power consumption and larger clock skew, clock gating can be adopted to reduce power consumption, in that the multiple meshes becomes more power efficient solution. But they did not use the actual clock gated circuits for their assessments, and the multi-level clock gating structure covered in our study was not also considered.

In [15], which is the preliminary version of this paper, Jung et al. in the first time consider practical multi-level clock gating structure in the design of the single-mesh and multiple-mesh network. They presented the comparison of the two mesh networks, and showed that the multiple-mesh network consumes lower power while the single-mesh has the advantages in clock skew and design complexity. It is also presented that the floorplanning of multiple meshes can be used to reduce the power consumption of multiple meshes at the cost of critical path delay.

8 Conclusion

The clock network of a design with hierarchical clock gating can be implemented by a set of meshes. If some hierarchies are removed, however, it also can be implemented by a single big mesh. We have shown that multiple-mesh implementation has advantage in clock power (16.2% smaller power on average of test circuits); but single mesh consumes shorter clock wires, yields smaller clock skew, and takes less time to design.

Multiple meshes can be floorplanned with some overlaps if placement is performed in flat, or they can be floorplanned without overlap if hierarchical physical design is assumed. The experiments have shown that the mesh floorplan with

overlap yields smaller clock power owing to shorter clock wires, smaller clock skew, and more variation tolerance, but timing closure is easier if overlap are not allowed.

References

1. Shin, Y., Shin, K., Kenkare, P., Kashyap, R., Lee, H.J., Seo, D., Millar, B., Kwon, Y., Iyengar, R., Kim, M.S., Chowdhury, A., Bae, S.I., Hong, I., Jeong, W., Lindner, A., Cho, U., Hawkins, K., Son, J.C., Hwang, S.H.: 28nm high-k metal-gate heterogeneous quad-core CPUs for high-performance and energy-efficient mobile application processor. In: Proceedings of International Solid-State Circuits Conference, pp. 154–155, February 2013
2. Singh, T., Bell, J., Southard, S.: Jaguar: a next-generation low-power x86–64 core. In: Proceedings of International Solid-State Circuits Conference, pp. 52–53, February 2013
3. Xu, K., Choy, C.S.: Low-power H.264/AVC baseline decoder for portable applications. In: Proceedings of International Symposium on Low Power Electronics and Design, pp. 256–261, August 2007
4. Guthaus, M.R., Wilke, G., Reis, R.: Revisiting automated physical synthesis of high-performance clock networks. *ACM Trans. Design Autom. Electron. Syst.* **18**(2), 31:1–31:27 (2013)
5. Chinnery, D.: High performance and low power design techniques for ASIC and custom in nanometer technologies. In: Proceeding of International Symposium on Physical Design, pp. 25–32, March 2013
6. Cyclos: Clock design for SoCs with lower power and better specs. <http://www.cyclos-semi.com>
7. Shim, S., Mo, M., Kim, S., Shin, Y.: Analysis and minimization of short-circuit current in mesh clock network. In: Proceedings of International Conference on Computer Design, pp. 459–462, October 2013
8. Ezroni, J.: Advanced dynamic power reduction techniques: XOR self-gating. White paper, April 2011
9. OpenCores. <http://www.opencores.org>
10. Rajaram, A., Pan, D.Z.: MeshWorks: a comprehensive framework for optimized clock mesh network synthesis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **29**(12), 1945–1958 (2010)
11. Lu, J., Mao, X., Taskin, B.: Integrated clock mesh synthesis with incremental register placement. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **31**(2), 217–227 (2012)
12. Guthaus, M.R., Wilke, G., Reis, R.: Non-uniform clock mesh optimization with linear programming buffer insertion. In: Proceedings of Design Automation Conference, pp. 74–79 (2010)
13. Lu, J., Mao, X., Taskin, B.: Clock mesh synthesis with gated local trees and activity driven register clustering. In: Proceedings of International Conference on Computer-Aided Design, pp. 691–697, April 2012
14. Wilke, G.R.: Design and analysis of “tree+local meshes” clock architecture. In: Proceedings of International Symposium on Quality Electronic Design, pp. 165–170, March 2007
15. Jung, J., Lee, D., Shin, Y.: Design and optimization of multiple-mesh clock network. In: Proceedings of International Conference on VLSI and System-on-Chip, pp. 171–176, October 2014