

Clock Mesh Network Design with Through-Silicon Vias in 3D Integrated Circuits

Kyungin Cho, Cheoljon Jang, and Jong-wha Chong

Many methodologies for clock mesh networks have been introduced for two-dimensional integrated circuit clock distribution networks, such as methods to reduce the total wirelength for power consumption and to reduce the clock skew variation through consideration of buffer placement and sizing. In this paper, we present a methodology for clock mesh to reduce both the clock skew and the total wirelength in three-dimensional integrated circuits. To reduce the total wirelength, we construct a smaller mesh size on a die where the clock source is not directly connected. We also insert through-silicon vias (TSVs) to distribute the clock signal using an effective clock TSV insertion algorithm, which can reduce the total wirelength on each die. The results of our proposed methods show that the total wirelength was reduced by 12.2%, the clock skew by 16.11%, and the clock skew variation by 11.74%, on average. These advantages are possible through increasing the buffer area by 2.49% on the benchmark circuits.

Keywords: 3D clock mesh distribution network, 3D IC design, clock TSV insertion, through-silicon via, TSV.

I. Introduction

The CMOS technology process is continuing to decline, and as the design of its integrated circuits becomes more complex, the capability to optimize performance is fast approaching the limit. Three-dimensional integrated circuits (3D ICs) using through-silicon vias (TSVs) have attracted considerable attention as an important technology; these are continuing the decreasing technology process trend that Moore predicted [1]. A 3D IC connects multiple dies that are vertically stacked on top of each other. The TSVs are used to connect the vertically stacked dies. A TSV is a via through the dies. This 3D stacking technology can significantly reduce mean and maximum wirelength; power consumption; chip area; and signal delay [2]–[3]. The methods of reducing the wirelength with TSVs in 3D ICs can also be applied to clock distribution networks [4]–[6].

A clock distribution network functions to propagate a clock signal from a clock source to clock sinks. The clock skew is defined as the maximum difference of a clock signal's arrival time from a clock source to all of the clock sinks. A high clock skew directly affects the maximum frequency and timing of circuits, which leads to a degradation of chip performance. According to the International Technology Roadmap for Semiconductors projection [7], the clock skew is generally required to be less than 3% to 4% of a clock period in a clock network design. To reduce the clock skew variation, many methods have been studied including variation-aware buffer and wire sizing [8], variation-aware routing [9], link insertion in clock trees [10], and leaf-level mesh [11]–[12]. Among these different methods to reduce clock skew variation, the leaf-level mesh with a top-level tree demonstrates highly effective results in several commercial chips [11]. The clock mesh network

Manuscript received Nov. 28, 2013; revised Apr. 30, 2014; accepted May 17, 2014.

This work was supported by the MSIP (Ministry of Science, ICT & Future Planning), Rep. of Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency), (NIPA-2013-H0301-13-1011).

Kyungin Cho (ruddls1116@gmail.com) and Jong-wha Chong (corresponding author, jchong@hanyang.ac.kr) are with the Department of Electronics Computer Engineering, Hanyang University, Seoul, Rep. of Korea.

Cheoljon Jang (jangcj@hanyang.ac.kr) is with the Department of Nanoscale Semiconductor Engineering, Hanyang University, Seoul, Rep. of Korea.

connects the clock sinks to the metal wires by intersecting vertical and horizontal metal wires. The clock signal is distributed from a clock source with a top-level tree to the clock sinks through the mesh driver located on the vertical and horizontal metal wire intersection. The clock mesh network requires more resources, including wirelength and power, than the clock tree network [11]. In addition, in a 3D clock mesh network, the wirelength is increased in proportion to the number of dies. For this reason, clock trees is more actively researched than clock mesh.

However, a clock mesh network guarantees a global skew variation, while a clock tree network guarantees a local skew variation. Well known for having a low global skew variation, the clock mesh network is mainly used in microprocessor design for high performance. With this in mind, researches on 3D clock mesh networks focus on reducing resources such as wirelength and power.

Because of the advantages of a clock mesh network, many studies related to mesh synthesis and optimization have been conducted [12]–[20]. The work of [13] proposed to remove only mesh wires that did not significantly impact on the clock skew of the mesh. The authors in [13] also suggest the method of buffer placement and sizing. The work of [14] aims at connecting clock sinks to mesh wires using a Steiner tree avoiding individually connecting clock sinks to mesh wires in an effort to reduce the stub wire of the mesh. A method is proposed in [12] to determine the buffer driver insertion and sizing and to remove the mesh wire for power consumption. In [15], the authors consider a timing delay on the path of the combinational circuits to determine the size of the initial mesh, by considering the timing when the clock mesh is constructed. A non-uniform clock mesh grid is proposed in [16]–[17] to reduce power consumption. A method is proposed in [18] to generate the clock mesh grid wires using an integer linear programming formulation to minimize the wirelength of the mesh. The works of [19] and [20] propose methods to simultaneously reduce the wires of a mesh grid and the stub wires of the mesh by placing the mesh grid wires close to the clock sinks. A method is proposed in [21]–[22] to determine the initial mesh size by considering the clock skew and wirelength. They also proposed a method for buffer placement, sizing, and wire sizing. In these works with 2D clock mesh networks, the clock signal is propagated to clock sinks in the x - and y -direction. However, in a 3D clock mesh network, the designs are more complicated because the clock signal is propagated to the clock sinks in three directions (x -, y -, and z -direction) using clock TSVs. In this sense, conventional 2D clock mesh network methods [12]–[22] did not consider the method of clock TSV insertion for propagating the clock signal on multiple dies. When a conventional 2D clock mesh network

is directly applied to a 3D IC by using clock TSVs regularly inserted at nodes of the mesh, the low skew variation is guaranteed but wirelength is increased. The wirelength is directly affected by the position and number of clock TSVs, and any unnecessary increases in wirelength can cause thermal issues on a 3D IC consisting of multiple vertically stacked dies. In addition, if a conventional 2D clock mesh network is directly applied to a 3D IC without using TSV insertion, then the reliability of the chip is decreased. Also, this method will not guarantee the global clock skew variation and will lead to a decrease in chip performance. For these reasons, it is necessary to study 3D ICs consisting of 3D clock mesh networks with clock TSVs.

In this work, we present an effective method to reduce the clock skew and wirelength using clock TSVs in a 3D clock mesh network.

The contributions of this paper are as follows:

- We propose a method to select the size of the mesh on each die. By inserting clock TSVs, the proposed mesh size selection algorithm has a lower clock skew than the clock skew constraint. Additionally, the mesh size selection algorithm can reduce the total clock mesh wirelength by decreasing the wirelength of the clock mesh grid.
- We present a method to insert clock TSVs close to clock sinks. The proposed clock TSV insertion and local mesh sizing methods achieve a low clock skew variation, although the size of the mesh is sparse. This is possible since the clock TSVs are inserted close to the clock sinks.
- We suggest a clock buffer assignment method to reduce the clock skew variation caused by an imbalanced stub wirelength. A proposed buffer assignment method can reduce the clock skew variation by inserting a buffer at regular intervals on the wires that go from the mesh driver to the clock sink. Additionally, the proposed buffer assignment method selects the number of buffers that can minimally reduce the clock skew.

The rest of the paper is organized as follows. In Section II, we discuss the preliminary research and background for the 2D clock mesh design. In Section III, the proposed 3D clock mesh design methodology is explained in detail. In Section IV, the simulation results are presented. The paper ends with concluding remarks in Section V.

II. Preliminary Research and Background of 2D Clock Mesh Network Design

The most important elements in clock mesh design are the wirelength and clock skew. In a clock mesh network, because of a wire's resistance and capacitance, wirelength affects power consumption. A high clock skew degrades the maximum

operating frequency of the chip and causes a signal timing issue. We present stub and mesh grid wires, which are affected by the mesh size in a 2D clock mesh network (see Section II-1). In Section II-2, clock skew and the size of the mesh in a 2D clock mesh network are discussed.

1. Wirelength in a 2D Clock Mesh Network

In a 2D clock mesh network, the total wirelength L_{total} is calculated as follows:

$$L_{total} = L_{top} + L_{mesh} + L_{stub}, \quad (1)$$

where L_{top} is the wirelength of the top-level tree, L_{mesh} is the wirelength of the mesh grid wire, and L_{stub} represents the stub wirelength of the stub wire, which is the wire connecting the mesh grid to the clock sink (see Fig. 1). The stub and mesh grid wires implicitly affect the power consumption of the circuit. As the wirelength increases, the entire capacitance and power consumption of the circuit are increased. Consequently, to reduce power consumption in the clock mesh network, we need to reduce both the length of the stub wires and the mesh grid wires. In this paper, to reduce L_{mesh} , we select the smallest possible mesh size such that the selected mesh has a lower clock skew than the clock skew constraint $t_{skew}^{constraint}$. We simply insert clock TSVs to reduce L_{stub} .

2. Clock Skew in 2D Clock Mesh Networks

In [13], the global clock skew t_{skew} is estimated as follows:

$$t_{skew} = t_{skew}^{buf} + D_{mesh}(d_{max}) + D_{stub}(L_{stub}^{max}), \quad (2)$$

where $t_{skew}^{constraint}$ is the skew introduced by the difference in the maximum and minimum delay on the mesh driver, $D_{mesh}(d_{max})$ is the maximum delay from the mesh drivers to the points where the stub wires meet with the mesh grid wires, and $D_{stub}(L_{stub}^{max})$ is the maximum delay from the points where the stub wires meet with the mesh grid wires to the clock sinks. In (2), $D_{mesh}(d_{max})$ and $D_{stub}(L_{stub}^{max})$ must be reduced to decrease the clock skew. As we move to the right in Fig. 1, the size of the mesh is increased and t_{skew} is reduced, because the

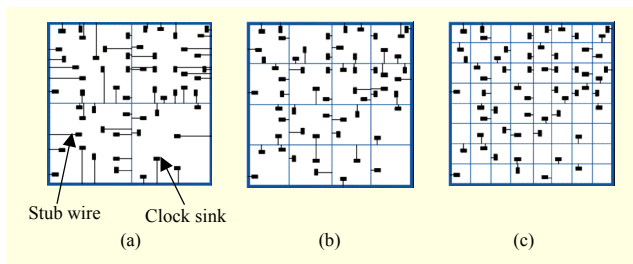


Fig. 1. Clock mesh size.

$D_{mesh}(d_{max})$ and $D_{stub}(L_{stub}^{max})$ are reduced. However, as the size of the mesh increases, the number of horizontal and vertical metal wires increases. Therefore, in this paper, we suggest a method to construct a clock mesh network that considers both L_{total} and t_{skew} .

III. Proposed Methodology

1. Overview

The flow of the proposed method is presented in Fig. 2. The first step is the selection of the mesh size. The uniform size of the mesh on each die is selected when the size of the mesh has both the minimal total wirelength and a lower clock skew than the clock skew constraint. After the size of the mesh is selected, the buffer-inserted top-level tree is constructed to simultaneously transfer the clock signal from a clock source to all of the clock sinks. A clock TSV at the center of die 1 is inserted to transfer the clock signal from die 1 to die 2 with minimum wirelength. After the initial mesh construction, clock TSVs are inserted and the local mesh-sizing algorithm is performed to reduce $D_{stub}(L_{stub}^{max})$. In the last step, the buffer assignment, the buffers are inserted to reduce the delay that occurs in the imbalanced L_{stub} .

2. Mesh Size Selection

As mentioned in Section II, the size of the mesh affects both the wirelength and the clock skew. In Fig. 1, it can be seen that the denser the clock mesh, the lower the value of L_{stub} . The global clock skew t_{skew} of a 3D IC is represented below

$$t_{skew} = t_{skew}^{buf} + D_{mesh}(d_{max}) + D_{stub}(L_{stub}^{max}) + D_{TSV}, \quad (3)$$

where the variable D_{TSV} is the delay caused by inserting the

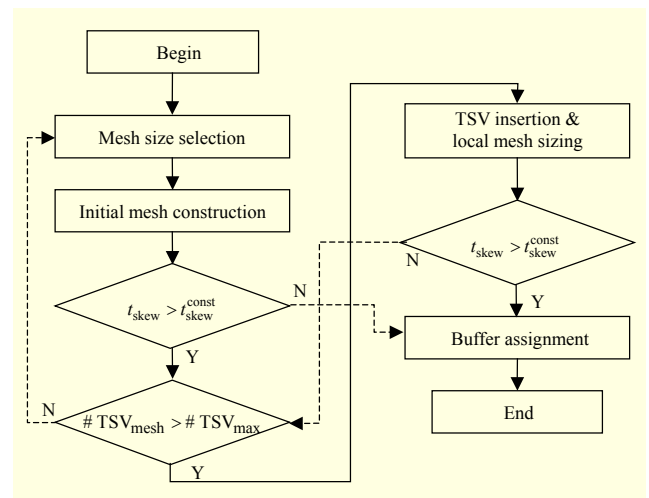


Fig. 2. Flow of proposed methodology.

Algorithm 1. Mesh size selection algorithm.Input: clock sink locations on each die, skew constraint, #TSV_{max}Output: mesh size with a minimum L_{total}

```

1: for each die  $i$  do
2:   initialize  $m, n = 1$ ;
3:   calculate the  $D_{sink \in die 1}$  and  $D_{sink \in die 2}$ ;
4:   while ( $t_{skew} > t_{skew}^{constraint}$ ) do
5:     if (# TSVmesh < # TSVmax)
6:       TSV insertion and local mesh sizing;
7:       Increment value of # TSVmesh by 1;
8:     else
9:        $m = m + 1$ ;
10:       $n = n + 1$ ;
11:      calculate the  $D_{sink \in die 1}$  and  $D_{sink \in die 2}$ ;
12:      if ( $t_{skew} < t_{skew}^{constraint}$ )
13:        return  $m, n$  stop;
14:      end while
15:   end for

```

Fig. 3. Pseudocode for mesh size selection procedure.

clock TSVs between die 1 and die 2. The total wirelength L_{total} is represented as

$$L_{total} = L_{top} + L_{mesh} + L_{stub} + L_{TSV}, \quad (4)$$

where the variable L_{TSV} is the total wirelength of the inserted clock TSVs. We assume that the all clock TSVs have the same wirelength. After the clock signal is instantaneously transferred to each mesh node, the mesh that has a lower L_{mesh} and L_{stub} also has a lower t_{skew} . However, through (4), the sparser mesh has a lower total wirelength than the denser mesh. Therefore, we need to select the size of the mesh by considering both the total wirelength and the clock skew.

There are many works that consider the above elements when selecting the mesh size in a two-dimensional integrated circuit. A method is proposed in [23] to select the minimum mesh size so as to reduce both the wirelength and the clock skew on each voltage domain. In [21], the constraints of the clock skew and wirelength are considered when the initial mesh is constructed so that the total wirelength is lower than L_{const} and the clock skew is lower than $t_{skew}^{constraint}$.

The proposed mesh size selection algorithm expands upon the initial mesh sizing method of [21] for a 3D clock mesh network with clock TSVs. To select the mesh size of minimal wirelength, we select a candidate mesh size and insert clock TSVs using our proposed method of clock TSV insertion. After the above procedure, we select the mesh size of minimum wirelength by calculating the clock skew. In this paper, we assume that the vertical metal wire m and horizontal metal wire n for the clock mesh are of the same value. In physical IC design, a uniform clock mesh is generally preferred since the mesh grid can be placed between uniform power rails to

prevent crosstalk [17]. The detailed proposed mesh size-selection algorithm is presented in Fig. 3.

The proposed mesh size-selection method selects the mesh size having minimum total wirelength when the clock skew of the selected mesh is lower than clock skew constraint. The inputs of the proposed mesh size-selection algorithm are the location of the clock sinks on each die, the clock skew constraint, and the maximum number of TSVs (TSV_{max}). We select the minimum candidate mesh size so as to obtain the minimum total wirelength (line 2 of Fig. 3). The delay from the clock source to the clock sinks on each die is calculated using (5) and (6) with the Elmore delay model (line 3 of Fig. 3). The delay $D_{sink \in die 1}$ from the clock source to the clock sink located on die 1, where it is directly connected with the clock source, is represented as (5) by Elmore delay modeling.

$$D_{sink \in die 1} = R_d (C_w + C_s) + R_w \left(\frac{C_w}{2} + C_s \right) + D_{sd} + D_d. \quad (5)$$

In (5), R_d and R_w are the resistance of the mesh driver and the resistance of the wire, respectively. The variables D_d , D_{sd} , C_w , and C_s are the intrinsic delay of the mesh driver, the delay from the clock source to the mesh driver, the capacitance of the wire, and the capacitance of the clock sink, respectively. The delay $D_{sink \in die 2}$ from the clock source to the clock sink located on die 2, where the clock signal is transferred from the clock source to the clock sink through the clock TSVs, is represented as (6) by Elmore delay modeling.

$$D_{sink \in die 2} = R_d (C_{TSV} + C_d) + R_{TSV} \left(\frac{C_{TSV}}{2} + C_{Tb} \right) + D_{Tb} + R_{Tb} (C_w + C_s) + R_w \left(\frac{C_w}{2} + C_s \right) + D_{sd} + D_d. \quad (6)$$

The variables R_{TSV} and R_{Tb} are the resistance of the clock TSV and the resistance of the clock TSV buffer, respectively. The variables C_{TSV} , C_d , C_{Tb} , and D_{Tb} are the capacitance of the clock TSV, mesh driver, clock TSV buffer, and the intrinsic delay of the clock TSV buffer, respectively. The delay models of the wire, clock TSV, and clock buffer are shown in Fig. 4.

To calculate the delay from the mesh driver on die 1 to the clock sinks on dies 1 and 2, we use the delay model illustrated in Fig. 4. If the calculated t_{skew} is higher than $t_{skew}^{constraint}$, then clock TSVs are inserted to reduce t_{skew} (lines 4–7). The insertion of the clock TSVs is performed using Algorithm 2 (introduced in Section III-4). If t_{skew} is higher than $t_{skew}^{constraint}$ after the maximum number of TSVs is inserted, then the above process is repeated by increasing the size of the mesh by one (lines 8–10). During this iteration, the size of the mesh is

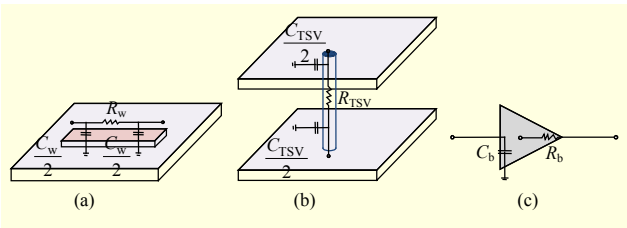


Fig. 4. Delay models of (a) wire, (b) clock TSV, and (c) clock buffer.

selected when the clock skew is less than clock skew constraint set by the designer (lines 12–13).

3. Initial Mesh Construction

After the candidate mesh size is selected, an initial clock TSV needs to be inserted to transfer the clock signal from die 1 to die 2. This is inserted at the center of die 1 so as to propagate the clock signal from die 1 to die 2 with minimum wirelength.

4. TSV Insertion and Local Mesh Size

In this step, clock TSVs are inserted on die 1 so that the clock signal is propagated to the clock sinks on die 2. We calculate the delay from the clock source to the clock sink with (5) and (6). To obtain a clock skew value that is less than the clock skew constraint, a clock TSV is inserted near to the clock sink having the largest delay from the clock source located in die 2. Figure 5 shows the clock TSV insertion process with a three-step search, and Fig. 6 shows the detailed procedure of the clock TSV insertion method.

We assume that the clock sink with the largest delay from the clock source — namely, the target sink — is located at (x_s, y_s, z_s) on die 2 (see Fig. 5(a)). The candidate location of a clock TSV (x_k, y_k, z_s) is determined so as to be on the same die as the target sink. The detailed procedure for inserting a clock TSV is as follows. The variables i and n are the number of steps and the number of horizontal (and also vertical) metal wires, respectively.

A. Step. 1 ($i = 1$)

We select the center of each mesh grid as a candidate location for the insertion of a clock TSV $((x_k, y_k, z_s)$, line 1). To insert a clock TSV close to the target sink (x_s, y_s, z_s) , the distance between the candidate locations and the location of the target sink are compared using (line 2)

$$\text{dist} = \|(x_s - x_k), (y_s - y_k), (z_s - z_k)\| \quad (7)$$

$$\{s, k\} \in N, N = \{1, 2, \dots, n^2\}.$$

As the distances between (x_k, y_k, z_s) and (x_s, y_s, z_s) decrease in

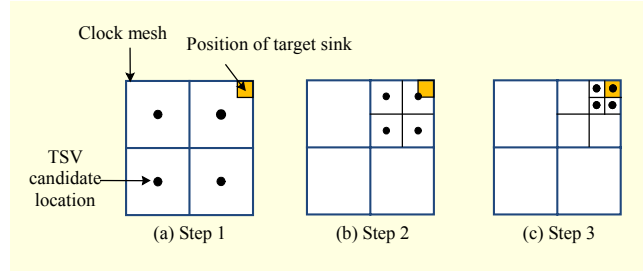


Fig. 5. Clock TSV insertion with three-step search on die 2.

Algorithm 2. TSV insertion and clock mesh sizing.

Input: clock sink locations in 3D (x_s, y_s, z_s) , mesh size of each die

Output: clock mesh network with TSVs

- 1: $(x_k, y_k, z_s) =$ center location of mesh grid N
- 2: calculate the dist by (7);
- 3: for $(i = 1, i < 4)$ do
- 4: update (x_k, y_k, z_s) to (x_i, y_i, z_s) with the shortest dist;
- 5: increment value of i by 1;
- 6: for $(t = 0, T < 4)$ do
- 7: calculate the dist by (7);
- 8: increment value of T by 1;
- 9: end for
- 10: end for
- 11: compare the dist from (x_k, y_k, z_s) to $(x_1, y_1, z_s), (x_2, y_2, z_s), (x_3, y_3, z_s)$
- 12: insert TSV at the location with minimized dist
- 13: increase the size of local mesh at TSV inserted position on (x_i, y_i, z_{s-1})

Fig. 6. Pseudocode for TSV insertion and local mesh sizing.

length, so the wirelength between the target sink and a clock TSV will also decrease. We select the candidate location (x_i, y_i, z_s) for the insertion of a clock TSV by using the lowest value from the results of (7) (line 4).

B. Step. 2 ($i = 2$)

The mesh grid that is selected in Section III-4-A, including a candidate location from the result of Section III-4-A, is bisected in both the vertical and horizontal directions by the candidate location. Each center location of the resulting four regions then becomes a new candidate location for a clock TSV (x_k, y_k, z_s) ; these TSVs can be inserted using

$$(x_k, y_k, z_k) = (x_{i-1}, x_{i-1}, z_s) - M_T \frac{1}{n} \left(\frac{x}{2^i}, \frac{y}{2^i}, 0 \right), \quad (8)$$

$$t_0 = (1, 1, 0),$$

$$t_1 = (1, -1, 0).$$

In (8), T is the set consisting of the center locations $t_0, t_1, t_2,$ and t_3 , of the four regions, divided in both the vertical and horizontal directions by the clock TSV location (x_{i-1}, y_{i-1}, z_s) . The location of the clock TSV (x_i, y_i, z_s) in Section III-4-B is the lowest value from the results of (7) (lines 7–8 and 4).

C. Step 3 ($i = 3$)

In this step, the process performed in Section III-4-B is repeated. Through these processes, the candidate location of the clock TSV(x_3, y_3, z_3) that is selected in this step may not be the closest location to the target sink. In other words, the distance from the target sink to the selected location in this step may not be closer than the same corresponding distances found in the previous steps. Therefore, to insert the clock TSV at the closest possible location to the target sink, the results of all three steps are compared (lines 11–12). The clock signal will be propagated to die 2 through a clock TSV from die 1 when the clock TSV is vertically inserted. After the clock TSV is connected to a mesh node on die 1, the clock signal is transferred simultaneously to both the mesh nodes, where the clock TSV is connected, and the other mesh nodes on die 2. However, the proposed clock TSV insertion algorithm may not situate the clock TSV at the position of the mesh node on die 1. To solve the above issue, we proposed a method to locally increase the size of the mesh on die 1. When the size of the mesh is locally increased, the mesh nodes are added. The clock TSV is inserted at the added mesh node on die 1 (line 13). The method for locally increasing the size of the mesh on die 1 is described in Fig. 7 below.

The vertical and horizontal mesh wires are generated at the clock TSV location on die 1 and expand in the vertical and horizontal directions of the points until the uniform mesh grid wires are connected. Figure 7(a) shows the location of a clock TSV determined in step 1, and Figs. 7(b) and 7(c) show the locally increased size of the mesh on die 1 when a clock TSV is inserted. Figure 8 shows an example of the increased size of mesh on die 1. The clock TSVs that are inserted close to the target sinks, by Algorithm 2, affect the delay from the clock source to not only the target sink but also the other clock sinks. Therefore, the clock skews are recalculated by using (5) and (6). The clock TSV insertion is repeated until the clock skew is less than the clock skew constraint. The iteration is completed when the maximum number of clock TSVs ($\# \text{TSV}_{\max}$) have been used.

Figure 8(a) shows the inserted clock TSVs on the clock

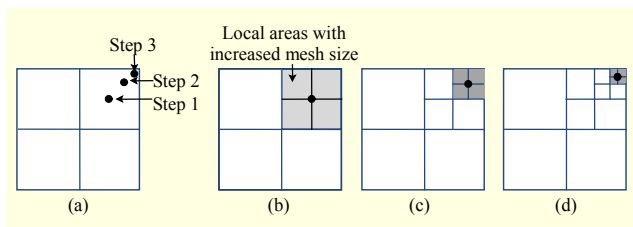


Fig. 7. Size of the mesh on die 1 is locally increased by inserting the clock TSV: (a) locations of a clock TSV as determined in steps 1–3, and (b)–(d) size of the mesh locally increased at die 1.

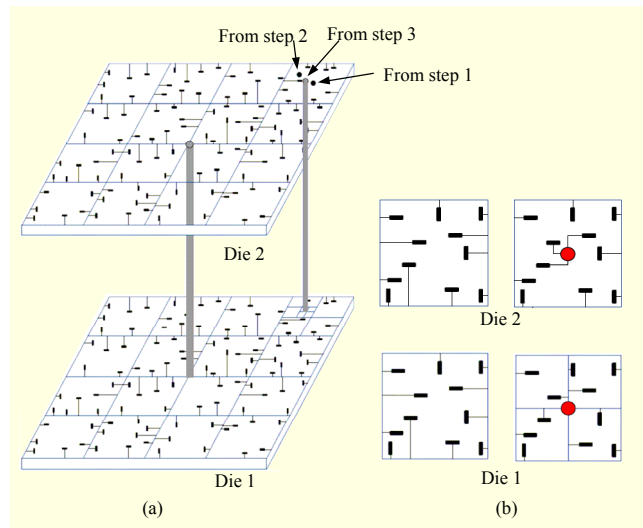


Fig. 8. 3D clock mesh with clock TSVs: (a) clock TSV insertion between die 1 and die 2, and (b) clock sinks connected to a clock TSV on die 2 and the locally increased size of the mesh at die 1.

mesh. Figure 8(b) shows an example of the clock TSV on die 2 that connects not only to the target sink but also to other clock sinks; they are connected at the clock TSV if the distance from the clock TSV is closer than the nearby mesh nodes.

5. Buffer Assignment

The stub wirelength on die 2 is relatively longer than that on die 1 because the size of the mesh on die 1 is larger than that of die 2. As the wirelength is increased, the signal delay from the clock source to the clock sinks is generally increased since both the capacitance and resistance of the wire are increased [24]. In this step, we propose a method to assign buffers to reduce the delay from the longer stub wirelength. The method to insert buffers to reduce the delay is well studied in [24]–[25]. The proposed method in [25] assumes that the distance from the source to the nearest buffer is x and that the distance between each buffer is y . However, the delay equations proposed in [25] are complex and have a lot of variables. In [24], the authors assume that the distance between each buffer is x ; therefore, the delay equations are more simple than in [25]. In this paper, we proposed a buffer assignment method to find the optimized number of buffers using the simple delay equation presented in [24], which can reduce the delay. We assume that all of the wire parameters, such as the capacitance per unit length and the resistance per unit length, are the same; although these parameters are in fact different. We assume that a number of buffers, k , are inserted between either the clock TSV on die 2 or the mesh driver on die 1 and the clock sink. To insert the buffers on the highly delayed wire, the buffers are selected in

the buffer library. To find k , the minimum delay D_m is obtained by

$$D_m = R_{Tb} (c_w x + C_b) + D_{Tb} + \frac{1}{2} r_w c_w x^2 + r_w x C_b + (k-1) \left[R_b (c_w x + C_b) + D_b + \frac{1}{2} r_w c_w x^2 + r_w c_w C_b \right] + R_b [c_w (L - kx) + C_s] + D_b + \frac{1}{2} r_w c_w (L - kx)^2 + r_w (L - kx) C_s. \quad (9)$$

In (9), R_{Tb} , R_b , r_w , and c_w are the resistance of the buffer with the clock TSV, the resistance of the buffer, the resistance per unit length of wire, and the capacitance per unit length of wire, respectively. The variables C_b , C_s , D_{Tb} , and D_b are the capacitance of the clock buffer, clock sink, the intrinsic delay of the buffer with the clock TSV, and clock buffer, respectively. We assume that the clock buffers are inserted at the position where the wirelength L from a clock TSV buffer to a clock sink is divided by $k + 1$. According to this assumption, the distance between each buffer is $x = (k + 1)^{-1}$. We assign x to (9), and the delay D_m is represented as follows:

$$D_m = (k + 1) \left[R_b \left(C_w \frac{1}{k + 1} + C_b \right) + D_b + \frac{r_w c_w L^2}{2(k + 1)^2} + \frac{r_w c_w L}{k + 1} \right]. \quad (10)$$

According to the method in [25], it is effective to insert a buffer when the delay from the buffer insertion is less than the delay without the buffer. This method is generalized as follows:

$$D_m (k - 1) > D_m (k). \quad (11)$$

Equation (10) is assigned to (11), and the result is represented as follows:

$$R_b C_b + D_b + \frac{r_w c_w L^2}{2(k + 1)} - \frac{r_w c_w L^2}{2k} < 0. \quad (12)$$

Rearranging (12) in terms of k we have

$$k^2 + k - \frac{r_w c_w L^2}{2(R_b C_b + D_b)} < 0. \quad (13)$$

Then, the solution of (13) is as follows:

$$k < \frac{-1 + \sqrt{1 + 2 \frac{r_w c_w L^2}{(R_b C_b + D_b)}}}{2}. \quad (14)$$

We are able to obtain the optimal number of buffers k with the minimum delay D_m by using (14). The clock skew t_{skew} is then calculated. If the calculated clock skew is greater than the clock skew constraint, then the above procedure is repeated for the highly delayed wire until the clock skew is lower than the clock skew constraint.

IV. Simulation Results

1. Design Environment

The algorithms were implemented in C++, and simulations were run on a Linux workstation with 2 GB of RAM. The proposed methods were verified using experiments performed on the ISCAS89 and ISPD 2010 benchmark circuits. The proposed method uses an existing placement result as the input. We compared our result with the result from the 2D clock mesh network in [21] and [22] because the clock mesh network in a 3D IC has not yet been presented. The study in [21] and [22] only constructed a 2D clock mesh network. We expand the results from [21] and [22] to a 3D clock mesh network. To do this, we stacked the same two dies vertically. The nominal skew constraint is 75 ps, which is the same as in [21]. We used 12 different buffer sizes with a maximum-capacitance limit ranging from 60 fF to 300 fF; this is the same as in [21]. We used buffer sizes with a maximum-capacitance limit ranging from 60 fF in the proposed Algorithm 3. The clock skew constraint used in Algorithms 1 and 2 is 50 ps. The clock skew constraint used in Algorithm 3 is 35 ps. We also used the same 65 nm technology parameters, transistor model from [21], and a similar set of benchmark circuits. In this paper, the resistance of the clock TSV is 0.053 Ω , and the capacitance of the clock TSV is 27.9 fF. The variation parameters considered are the buffer channel lengths, power supply variation, and sink load capacitance variation. These parameters are varied with a 5% standard deviation from their nominal value. We model the effects of the variation in the top-level tree in a similar way as [21] by modeling the input arrival time for the mesh drivers with a random variable. We used a range of ± 25 ps for the clock skew between two mesh drivers and used the same slew for all of the mesh buffers. The methods used in the simulation are presented as follows:

- In [21] and [22], the clock mesh was only constructed as a 2D clock mesh network. Therefore, we expanded the method from [21] and [22] to a 3D clock mesh network to compare our proposed TSV insertion, local mesh sizing, and buffer assignment method. We construct a 3D clock mesh by vertically stacking two identical dies that have a 2D clock mesh; this is proposed in [21] and [22]. To connect the two dies, the same number of clock TSVs were used in the proposed method and were inserted at regular space intervals. This approach is denoted by “[21]_EX” and “[22]_EX” in our tables.

- We construct a 3D clock mesh by applying the proposed Algorithm 3 to [21]_EX to demonstrate the effects of our proposed buffer assignment algorithm. This approach is denoted by “[21]_EX with buffer” in our tables.

- We run our 3D clock mesh algorithm on the clock mesh

Table 1. Comparison of clock mesh with maximum skew of ± 50 ps between mesh buffers and a slew of 50 ± 10 ps on ISCAS benchmarks.

Benchmark (# sinks)	Method	BA		WL		μ_{skew}	σ_{skew}
		μm^2	Ratio	μm	Ratio	ps	ps
S5378 (165)	[21]_EX	63.2	1.29	65382	1.14	31.8	8.7
	[21]_EX with buffer	69.6	1.42	65382	1.14	28.7	6.1
	[22]_EX	51.4	1.05	65382	1.14	30.1	7.4
	Proposal	48.7	1.00	57040	1.00	29.3	6.9
S13207 (500)	[21]_EX	168.5	1.10	245858	1.16	19	4.5
	[21]_EX with buffer	177.4	1.16	245858	1.16	17.1	3.8
	[22]_EX	152.6	0.99	245858	1.16	18.1	4.5
	Proposal	152.5	1.00	210311	1.00	17.9	4.6
S15850 (566)	[21]_EX	200.5	1.06	218550	1.23	19.2	4.0
	[21]_EX with buffer	214.1	1.13	218550	1.23	17.2	3.1
	[22]_EX	189.1	1.01	218550	1.23	17.8	3.8
	Proposal	188.7	1.00	176445	1.00	17.8	3.5
S35932 (1426)	[21]_EX	536.2	1.05	637424	1.13	23.7	5.1
	[21]_EX with buffer	588.2	1.15	637424	1.13	21.0	3.0
	[22]_EX	510.5	1.01	637424	1.13	21.9	3.6
	Proposal	509.4	1.00	559993	1.00	21.6	3.4
S38584 (1728)	[21]_EX	658.4	1.02	761346	1.10	28.7	4.9
	[21]_EX with buffer	742.0	1.15	761346	1.10	24.4	2.9
	[22]_EX	646.6	1.01	761346	1.10	26.9	4.4
	Proposal	640.6	1.00	689531	1.00	25.6	3.6

obtained from the clock mesh sizing, clock TSV insertion, and buffer assignment. This approach is denoted by “Proposal” in our tables.

2. Results

Table 1 shows the results of the 3D clock mesh network with the maximum clock skew of ± 50 ps between mesh drivers and a slew of 50 ± 10 ps on ISCAS benchmarks. The parameters in Table 1 are “BA” for the buffer area, “WL” for the total wirelength, “ μ_{skew} ” for the mean deviation of the clock skew, and “ σ_{skew} ” for the standard deviation of clock skew. The parameters μ_{skew} and σ_{skew} are obtained through a Monte Carlo simulation in HSPICE. The columns under “Ratio” are the relative value with respect to “Proposal.” For the ISCAS benchmarks, the proposed buffer assignment method, denoted by “[21]_EX with buffer,” shows the clock skew results reduced by 11.31% and the clock skew variation reduced by 30.02%, on average, compared to the method in [21]. However, the buffer area is generally increased when the clock buffers are inserted. Our proposed buffer assignment method increases the

buffer area by 8.92%, on average, by inserting the buffers.

The method with the iterative deletion of the buffer, which is proposed in [22], shows better performance than our buffer insertion algorithm. However, our proposed method with Algorithms 1 and 2, as well as the buffer assignment, shows a better performance in terms of buffer area, wirelength, and clock skew variation.

Table 2 shows the results of the 3D clock mesh network on ISPD 2010 benchmarks. Our proposed algorithm shows the clock skew results reduced by 4.2% and the wirelength reduced by 12.2%, on average, compared to the method in [22], which is expanded to the 3D clock mesh network. Table 3 shows the effects of variation on the top-level tree by modeling the input arrival time for the mesh drivers. The different parameter in Table 3 is the skew variation, which represents the clock skew difference between the mesh drivers. We obtained the reduced total wirelength and clock skew shown in Table 1 and Table 2. These results show the effectiveness of our proposed methods. When we assign the variation of skew to the buffers constructing the top-level tree, the buffer area is increased by approximately 2.48%, on average, more than the

Table 2. Comparison of clock mesh with maximum skew of ± 50 ps between mesh buffers and a slew of 50 ± 10 ps on ISPD 2010 benchmarks.

Benchmark (# sinks)	Method	BA		WL		μ_{skew}	σ_{skew}
		μm^2	Ratio	μm	Ratio	ps	ps
01	[21]_EX	1658.4	1.38	3232546	1.22	29.3	6.9
	[21]_EX with buffer	1891.2	1.58	3232546	1.22	26.1	3.1
	[22]_EX	1108.4	0.92	3232546	1.22	28.4	5.0
	Proposal	1193.2	1.00	2634824	1.00	27.9	4.5
02	[21]_EX	2832.5	1.47	3024654	1.18	42.1	9.8
	[21]_EX with buffer	3521.4	1.83	3024654	1.18	38.4	6.1
	[22]_EX	1953.4	1.01	3024654	1.18	40.6	7.3
	Proposal	1923.7	1.00	2542389	1.00	39.8	6.9
03	[21]_EX	953.4	1.23	2032485	1.25	19.9	9.6
	[21]_EX with buffer	990.7	1.28	2032485	1.25	14.2	4.4
	[22]_EX	753.4	0.97	2032485	1.25	16.1	6.9
	Proposal	769.4	1.00	1624685	1.00	15.8	5.9
04	[21]_EX	1035.2	1.18	2498327	1.27	15.9	8.9
	[21]_EX with buffer	1142.9	1.30	2498327	1.27	11.6	4.5
	[22]_EX	890.6	1.02	2498327	1.27	13.4	5.0
	Proposal	872.6	1.00	1954654	1.00	12.8	4.9
05	[21]_EX	893.5	1.21	2456872	1.34	16.7	9.4
	[21]_EX with buffer	953.4	1.30	2456872	1.34	13.1	3.8
	[22]_EX	742.8	1.01	2456872	1.34	15.8	6.4
	Proposal	732.5	1.00	1824648	1.34	14.8	5.8

Table 3. Results of different skew values for mesh buffer input signals.

Skew variation (ps)	Method	BA %Red	WL %Red	μ_{skew} AVG.	σ_{skew} AVG.
± 10	[21]_EX	0.00	0.00	26.88	8.42
	[21]_EX with buffer	-10.49	0.00	21.89	5.91
	[22]_EX	3.54	0.00	24.04	7.31
	Proposal	-2.49	12.2	23.03	7.29
± 30	[21]_EX	0.00	0.00	25.87	6.24
	[21]_EX with buffer	-10.49	0.00	21.97	5.07
	[22]_EX	8.65	0.00	24.18	5.98
	Proposal	-2.49	12.2	23.17	5.74
± 50	[21]_EX	0.00	0.00	27.45	5.45
	[21]_EX with buffer	-10.49	0.00	21.09	4.96
	[22]_EX	7.68	0.00	23.99	5.35
	Proposal	-2.49	12.2	22.84	5.11
Average	[21]_EX	0.00	0.00	26.73	6.70
	[21]_EX with buffer	-10.49	0.00	21.65	5.31
	[22]_EX	12.21	0.00	23.98	6.27
	Proposal	-2.49	12.2	23.79	6.05
Improvement	Proposal	-2.49	12.2	4.61	0.65

conventional method in [21] that was expanded to 3D. However, instead of a buffer area increased by 2.48%, our methods reduce the total wirelength by 12.2%, the clock skew by 16.11%, and the clock skew variation by 11.74%, on average.

V. Conclusion

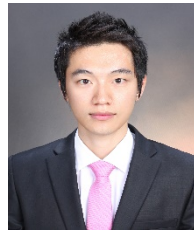
In this paper, we presented effective methods, used in the construction of a 3D clock mesh network, for the selection of mesh size, TSV insertion, local mesh sizing, and buffer assignment. From the simulation results, we verified that our proposed mesh size selection, clock TSV insertion, and local mesh sizing methods can reduce the total wirelength and that the proposed buffer assignment method can reduce the clock skew and clock skew variation. The simulation results show that the total wirelength is reduced by 12.2%, the clock skew by 16.11%, and the clock skew variation by 11.74%; this is compared with the conventional method in [21] that was expanded to 3D with the same number of clock TSVs as used in the proposed methods. These advantages are possible with the buffer area increased by 2.49% on the benchmark circuits. The above results show that our proposed method can construct an effective and powerful 3D clock mesh network.

References

- [1] Y.F. Tsai et al., "Three-Dimensional Cache Design Exploration Using 3DCacti," *Proc. IEEE Int. Conf. Comput. Des.: VLSI Comput. Processors*, San Jose, CA, USA, Oct. 2–5, 2005, pp. 519–524.
- [2] X. Dong and Y. Xie, "System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs)," *Asia South Pacific Des. Autom. Conf.*, Yokohama, Japan, Jan. 19–22, 2009, pp. 234–241.
- [3] Y. Deng and W. Maly, "A Feasibility Study of 2.5D System Integration," *Proc. IEEE Custom Integr. Circuits Conf.*, San Jose, CA, USA, Sept. 21–24, 2003, pp. 667–670.
- [4] X. Zhao, J. Minz, and S.K. Lim, "Low-Power and Reliable Clock Network Design for Through-Silicon Via (TSV) Based 3D ICs," *IEEE Trans. Compon., Packaging Manuf. Technol.*, vol. 1, no. 2, Feb. 2011, pp. 247–259.
- [5] X. Zhao and S.K. Lim, "Power and Slew-Aware Clock Network Design for Through-Silicon-Via (TSV) Based 3D ICs," *Asia South Pacific Des. Autom. Conf.*, Taipei, Taiwan, Jan. 18–21, 2010, pp. 175–180.
- [6] T.Y. Kim and T.W. Kim, "Clock Tree Embedding for 3D ICs," *Asia South Pacific Des. Autom. Conf.*, Taipei, Taiwan, Jan. 18–21, 2010, pp. 486–491.
- [7] International Technology Roadmap for Semiconductors (ITRS). Accessed Nov. 21, 2013. <http://www.itrs.net>
- [8] M.R. Guthaus, D. Sylvester, and R.B. Brown, "Clock Buffer and Wire Sizing Using Sequential Programming," *ACM/IEEE, Des. Autom. Conf.*, San Francisco, CA, USA, July 24–28, 2006, pp. 1041–1046.
- [9] L. Xiao et al., "Local Clock Skew Minimization Using Blockage-Aware Mixed Tree-Mesh Clock Network," *IEEE/ACM Int. Conf. Comput.-Aided Des.*, San Jose, CA, USA, Nov. 7–11, 2010, pp. 458–462.
- [10] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing Clock Skew Variability via Crosslinks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 6, June 2006, pp. 1176–1182.
- [11] P.J. Restle et al., "A Clock Distribution Network for Microprocessors," *Symp. VLSI Circuits, Dig. Techn. Paper*, Honolulu, HI, USA, June 15–17, 2000, pp. 184–187.
- [12] G. Venkataraman et al., "Combinatorial Algorithms for Fast Clock Mesh Optimization," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no. 1, Jan. 2010, pp. 131–141.
- [13] A. Rajaram and D.Z. Pan, "MeshWorks: An Efficient Framework for Planning, Synthesis and Optimization of Clock Mesh Network," *Asia South Pacific Des. Autom. Conf.*, Seoul, Rep. of Korea, Mar. 21–24, 2008, pp. 250–257.
- [14] R.S. Shelar, "An Algorithm for Routing with Capacitance/Distance Constraints for Clock Distribution in Microprocessors," *Int. Symp. Physical Des.*, San Diego, CA, USA, Mar. 29–Apr. 1, 2009, pp. 141–148.
- [15] A. Abdelhadi et al., "Timing-Driven Variation-Aware Nonuniform Clock Mesh Synthesis," *Proc. Symp. Great Lakes Symp. VLSI*, Providence, RI, USA, May 16–18, 2010, pp. 15–20.
- [16] M.R. Guthaus, G. Wilke, and R. Reis, "Non-uniform Clock Mesh Optimization with Linear Programming Buffer Insertion," *ACM/IEEE Des. Autom. Conf.*, Anaheim, CA, USA, June 13–18, 2010, pp. 74–79.
- [17] J. Lu, X. Mao, and B. Taskin, "Integrated Clock Mesh Synthesis with Incremental Register Placement," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 2, Feb. 2012, pp. 217–227.
- [18] M. Cho, D.Z. Pan, and R. Puri, "Novel Binary Linear Programming for High Performance Clock Mesh Synthesis," *IEEE/ACM Int. Conf. Comput.-Aided Des.*, San Jose, CA, USA, Nov. 7–11, 2010, pp. 438–443.
- [19] J. Lu, X. Mao, and B. Taskin, "Timing Slack Aware Incremental Register Placement with Non-uniform Grid Generation for Clock Mesh Synthesis," *Proc. Int. Symp. Physical Des.*, Santa Barbara, CA, USA, Mar. 27–30, 2011, pp. 131–138.
- [20] J. Lu, Y. Aksehir, and B. Taskin, "Register On MESH (ROME): A Novel Approach for Clock Mesh Network Synthesis," *IEEE Int. Symp. Circuits Syst.*, Rio de Janeiro, Brazil, May 15–18, 2011, pp. 1219–1222.
- [21] A. Rajaram and D.Z. Pan, "MeshWorks: A Comprehensive Framework for Optimized Clock Mesh Network Synthesis," *IEEE*

Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 29, no. 12, Dec. 2010, pp. 1945–1958.

- [22] M.R. Guthaus et al., “High-Performance Clock Mesh Optimization,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 17, no. 3, June 2012, p. 33.
- [23] C. Sitik and B. Taskin, “Multi-voltage Domain Clock Mesh Design,” *IEEE Int. Conf. Comput. Des.*, Montreal, Canada, Sept. 30–Oct. 3, 2012, pp. 201–206.
- [24] M. You and H. Shin, “Improvement of Delay and Noise Characteristics by Buffer Insertion,” *IEEK*, vol. 41, no. 6, June 2004, pp. 81–90.
- [25] C. Alpert and A. Devgan, “Wire Segmenting for Improved Buffer Insertion,” *Proc. Des. Autom. Conf.*, Anaheim, CA, USA, June 9–13, 1997, pp. 588–593.



Kyungin Cho received his BS degree in electronics engineering from the school of Electronics Computer Engineering, Inha University, Incheon, Rep. of Korea, in 2010. From 2010 to 2012, he was a researcher at the HE Department of LG Electronics, Pyeongtaek, Rep. of Korea. Since 2013, he has been with the Department of Electronics Computer Engineering, Hanyang University, Seoul, Rep. of Korea, where he is currently pursuing his MS degree. His main research interest is SoC design methodology, including the physical design and automation of 3D ICs.



Cheoljon Jang received his BS and MS degrees in electronics and computer engineering from the school of Electronics Computer Engineering, Hanyang University, Seoul, Rep. of Korea, in 2011 and 2013, respectively. He is currently pursuing his PhD degree in nanoscale semiconductor engineering at Hanyang University. His main research interest is SoC design methodology, including the physical design and automation of 3D ICs.



Jong-wha Chong received his BS and MS degrees in electronics engineering from Hanyang University, Seoul, Rep. of Korea, in 1975 and 1979, respectively and his PhD degree in electronics and communication engineering from Waseda University, Shinjuku-ku, Tokyo, Japan, in 1981. Since 1981, he has been a professor of the Department of Electronics Engineering, Hanyang University. From 1979 to 1980, he was a researcher at the C&C Research Center of Nippon Electronics Company, Shiba Minato, Tokyo, Japan. From 1983 to 1984, he was a visiting researcher at the Korean Institute of Electronics & Technology, Seongnam, Rep. of Korea. Between 1986 and 2008, he was a visiting professor at the University of California, Berkeley, USA. He was the chairman of the CAD & VLSI society of the Institute of the Electronic Engineers of Korea in 1993. In 2007, he was the president of the IEEK, and from 2009 to 2010, he was the president of the KIEEE. He is currently the chairman of the Fusion SoC Forum. His main research interests are SoC design methodology; including memory centric design and the physical design and automation of 3D ICs; indoor wireless communication SoC design for ranging and location; video systems; and power IT systems.