

Hybrid Structured Clock Network Construction¹

Haihua Su Sachin S. Sapatnekar

Department of Electrical and Computer Engineering
University of Minnesota, 200 Union St. SE
Minneapolis, MN 55455

Abstract

This paper hierarchically constructs a hybrid mesh/tree clock network structure consisting of overlying zero-skew clock meshes, with underlying zero-skew clock trees originating from the mesh nodes. We propose a mesh construction procedure, which guarantees zero skew under the Elmore delay model, using a simple and efficient linear programming formulation. Buffers are inserted to reduce the transition time (or rise time). As a post-processing step, wire width optimization under an accurate higher-order delay metric is performed to further minimize the transition time and propagation delay/skew. Experimental results show that the hybrid mesh/tree construction scheme can provide smaller propagation delay and transition time than a comparable clock tree.

1. Introduction

Clock distribution has become an increasingly challenging problem for VLSI designs, and careful design of clock networks is essential in high-performance circuits. The most common objectives of clock network design are to ensure near-zero skew, sharp edges and the optimal use of routing resources. In addition, it is also important to reduce the total propagation delay through the network.

Traditionally, clock networks have been designed as tree structures since these are considered to be easier to optimize, and various algorithms [2, 17] for constructing zero-skew clock trees under the Elmore delay metric have been explored in the past. Several delay/skew minimization techniques [8, 9] and the wire width optimization techniques [12, 13] have been proposed. As compared to tree structures, clock meshes are known to potentially yield better performance, but are seldom used since the reliable optimization of meshes is considered to be a difficult problem. Clock mesh techniques have been used before, for example, in [5] and [14], but unlike our approach, are most commonly used at the lowest level.

In this paper, we develop a novel technique that describes how some classes of mesh can be optimized very easily for zero skew using a simple linear-programming based formulation. An important property of our algorithm is that, like Tsay's method [17], it works in a bottom-up fashion in constructing the clock network, solving a computationally inexpensive subproblem at each intermediate step. An additional feature of this work is that it can be extended very easily to build clock trees with prespecified nonzero skews for cases when deliberate skews are used in the clock network for cycle borrowing [6].

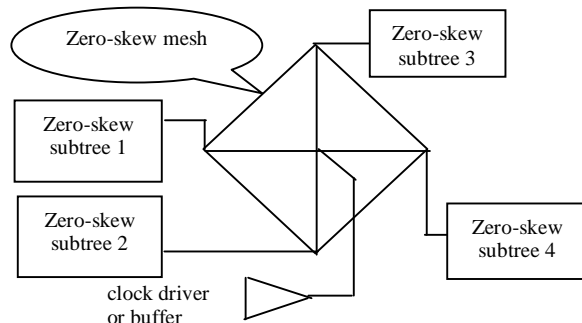


Fig. 1 A clock network as a mesh/tree structure

To blend the advantages of meshes and trees, we propose a hybrid mesh/tree structure that is guaranteed to have zero skew under the Elmore delay metric, and further optimize this structure to a target transition time and small non-zero skew under a higher order delay model. One example of our structured topology is a “one-level mesh” consisting of a global mesh feeding local trees, as shown in Fig. 1. This work has been inspired by a similar idea of using a topology that is intermediate to the two extremes of full trees and full meshes in [15, 16] for designing reliable power/ground networks.

In designing clock trees, “zero skew” is something of a misnomer [13], since routing constraints and process variations make non-zero skew inevitable, so that a more realistic objective is to construct reliable low-skew clock networks. For our constructed mesh/tree structure, a hierarchical simulator (HPRIMA) similar to [16] is used to find the higher order propagation delays and transition times at the clock pins. A reduced order modeling approach based on HPRIMA is also employed in calculating the adjoint sensitivities of delays, as well as of skew and transition time with respect to all the wire widths to guide our TILOS-like [7] heuristic wire width optimization for meeting more stringent skew and transition time constraints.

2. Zero-skew mesh construction

Our proposed clock network has a structure of overlying mesh with local trees rooted on the mesh, as shown in Fig. 1. Given a set of sinks, the construction is in a two-step bottom-up manner:

- (i) construct the *underlying zero-skew trees*; and
- (ii) construct the *overlying zero-skew mesh*.

We will not focus on step (i), since the zero-skew trees can be constructed using any of the conventional clock routing algorithms. Once these trees have been built, step (ii) requires the total delay and downstream capacitance for each such tree.

¹ This work was supported in part by the SRC under contract 609.001 and 884.001 and by the NSF under contract CCR-9800992.

For a one-level mesh/tree structure, we divide the sinks into four groups and build four trees, one for each group. Multi-level meshes can be built in a similar bottom-up way as the construction of the trees: at each level, each node in the i^{th} level mesh is connected to a tree whose sink nodes are $(i-1)^{\text{th}}$ or lower level meshes.

We will now illustrate the construction of a one-level zero-skew mesh shown in Fig. 2, consisting of the set of segments $\mathbf{S} = \{ab, bc, cd, ad, as, bs, cs, ds\}$. We assume that the four underlying trees have been built and that they are rooted at nodes a, b, c and d . Tree k has a delay of Δk and a downstream capacitor C_{Dk} , where $k = a, b, c$ and d . In our implementation, the (x, y) coordinates of the tapping point s of the mesh are chosen as the mean of corresponding coordinates of the four roots, but the succeeding derivation does not depend on this assumption.

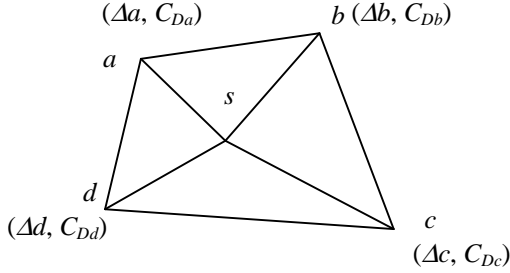


Fig. 2 One-level mesh with underlying trees

Each wire on the mesh is modeled as a segment under the π -model, with each segment modeled using lumped RC parameters given by

$$\begin{aligned} g_s &= w_s / \rho l_s \\ c_s &= \beta l_s w_s \end{aligned} \quad (1)$$

where l_s and w_s are the length and the width of the segment $s \in \mathbf{S}$, and the parameters ρ and β are the per square resistance and per square capacitance of the metal layer for the clock network. The admittance matrix [3] of the circuit is

$$G = \begin{bmatrix} g_{ab} + g_{ad} + g_{as} & -g_{ab} & 0 & -g_{ad} \\ -g_{ab} & g_{ab} + g_{bc} + g_{bs} & -g_{bc} & 0 \\ 0 & -g_{bc} & g_{bc} + g_{cd} + g_{cs} & -g_{cd} \\ -g_{ad} & 0 & -g_{cd} & g_{ad} + g_{cd} + g_{ds} \end{bmatrix} \quad (2)$$

and the capacitance matrix is

$$C = [C_a + C_{Da} \quad C_b + C_{Db} \quad C_c + C_{Dc} \quad C_d + C_{Dd}]^T \quad (3)$$

where C_k is the total wire capacitance at node k , $k = a, b, c$ and d .

The first-order (Elmore) delays from node s down to all the four roots are calculated by

$$\vec{\tau} = G^{-1}C \quad (4)$$

To achieve zero-skew, we require

$$\vec{\tau} = [D - \Delta a \quad D - \Delta b \quad D - \Delta c \quad D - \Delta d]^T \quad (5)$$

where D is the target delay from the tapping point s to all the sinks. Rewriting (4) as $G \vec{\tau} = C$, and substituting (1), (2), (3) and (5) into this new form, and let

$$\vec{w} = [w_{ab} \quad w_{bc} \quad w_{cd} \quad w_{ad} \quad w_{as} \quad w_{bs} \quad w_{cs} \quad w_{ds}]^T,$$

we have the following relation:

$$[A \mid B] \vec{w} = C \quad (6)$$

where

$$A = \begin{bmatrix} -\frac{\Delta_{ab}}{\rho l_{ab}} - \beta \frac{l_{ab}}{2} & 0 & 0 & -\frac{\Delta_{ad}}{\rho l_{ad}} - \beta \frac{l_{ad}}{2} \\ \frac{\Delta_{ba}}{\rho l_{ab}} - \beta \frac{l_{ab}}{2} & -\frac{\Delta_{bc}}{\rho l_{bc}} - \beta \frac{l_{bc}}{2} & 0 & 0 \\ 0 & -\frac{\Delta_{cb}}{\rho l_{bc}} - \beta \frac{l_{bc}}{2} & -\frac{\Delta_{cd}}{\rho l_{cd}} - \beta \frac{l_{cd}}{2} & 0 \\ 0 & 0 & -\frac{\Delta_{dc}}{\rho l_{cd}} - \beta \frac{l_{cd}}{2} & -\frac{\Delta_{da}}{\rho l_{ad}} - \beta \frac{l_{ad}}{2} \end{bmatrix} \quad (7)$$

and

$$B = \begin{bmatrix} L_1(D) & 0 & 0 & 0 \\ 0 & L_2(D) & 0 & 0 \\ 0 & 0 & L_3(D) & 0 \\ 0 & 0 & 0 & L_4(D) \end{bmatrix} \quad (8)$$

where $\Delta_{ij} = \Delta_i - \Delta_j$ and $L_i(D) = (D - \Delta_i) / (\rho l_{is}) - \beta l_{is} / 2$, $i, j = a, b, c$ and d .

Corresponding to matrices A and B , we also split \vec{w} into the upper half vector $[w_{ab} \quad w_{bc} \quad w_{cd} \quad w_{ad}]^T$ and the lower half vector $[w_{as} \quad w_{bs} \quad w_{cs} \quad w_{ds}]^T$. We can then rewrite the equation as

$$[w_{as} \quad w_{bs} \quad w_{cs} \quad w_{ds}]^T = B^{-1}(C - A[w_{ab} \quad w_{bc} \quad w_{cd} \quad w_{ad}]^T) \quad (9)$$

To achieve zero-skew, equation (9) must hold, and this implies that $w_{as} \sim w_{ds}$ can be expressed as a linear combination of $w_{ab} \sim w_{ad}$.

The linear programming model that creates a zero skew mesh can be formulated as follows:

Minimize Total wire area =

$$\begin{bmatrix} l_{ab} \\ l_{bc} \\ l_{cd} \\ l_{ad} \end{bmatrix} \bullet \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix} + \begin{bmatrix} l_{as} \\ l_{bs} \\ l_{cs} \\ l_{ds} \end{bmatrix} \bullet [B^{-1}(\vec{C} - A \begin{bmatrix} w_{ab} \\ w_{bc} \\ w_{cd} \\ w_{ad} \end{bmatrix})] \quad (10)$$

Subject to $\vec{w}_{\max} \geq B^{-1}(\vec{C} - A[w_{ab} \quad w_{bc} \quad w_{cd} \quad w_{ad}]^T) \geq \vec{w}_{\min}$ (11)

and $\vec{w}_{\max} \geq [w_{ab} \quad w_{bc} \quad w_{cd} \quad w_{ad}]^T \geq \vec{w}_{\min}$ (12)

where w_{ab} , w_{bc} , w_{cd} and w_{ad} are the decision variables for this problem, “ \bullet ” refers to the dot product of two vectors, and \vec{w}_{\min} and \vec{w}_{\max} are the lower-bounds and upper-bounds for wire widths which are positive technology dependent parameters. We note that the size of the linear program, in terms of both the number of variables and the number of constraints, is very small, so that its solution can be found very easily.

In order for the linear program to have a feasible solution, we now examine equation (6). We can always choose the target delay D such that each entry $(L_i(D))$, $i = a, b, c$ and d in matrix B is positive. A direct observation is that to make $w_{as} \sim w_{ds}$ greater than w_{\min} , we can make the diagonal entries in B smaller by either decreasing D or elongating l_{is} , $i = a, b, c$ and d . We still need to ensure that w_{ab} , w_{bc} , w_{cd} and w_{ad} lie above w_{\min} . Since the minimum wire width restriction is a hard limit imposed by the technology, we use a necessary condition to enforce this limit. It can be shown after some algebra, based on the facts that B is a diagonal positive definite matrix and that every entry in C is greater than zero, that the following is a necessary condition to ensure that the wire width does not violate the minimum width requirements:

$$A[w_{ab} \quad w_{bc} \quad w_{cd} \quad w_{ad}]^T \leq -B\vec{w}_{\min} \quad (13)$$

When any entry in matrix A is positive, it is possible to violate condition (13) and send the solution outside the feasible region. In such a situation, if we manage to make the LHS of (13)

smaller and the RHS of (13) larger, we will be able to draw the optimal solution back to the feasible region. This may be achieved through a wire elongation heuristic, which is stated as follows. Whenever there is a negative entry a_{ij} ($i, j =$ valid combinations of a, b, c or d) in matrix A , elongate the wire length l_{ij} to make the LHS of (13) smaller and elongate the wire length l_{is} to the corresponding entry in matrix B larger. Notice here that when we elongate l_{ij} , another entry in matrix A is also decreased, which makes this entry more negative, and therefore does not affect the feasibility of the corresponding constraint.

After wire elongation, if the linear programming solver still cannot find an optimal solution, it means that our target delay D has been chosen to be too strict. We must then relax the target delay to a reasonable value.

Our linear-programming based zero-skew mesh construction algorithm is as follows:

1. Choose a target delay D . Starting with the initial $D = \max\{\Delta a, \Delta b, \Delta c, \Delta d\}$, update the value of D by multiplying it with a small factor until all $L_i(D)$'s are positive, $i = a, b, c$ and d .
2. Check each entry in matrix A : if any entry is positive, heuristically elongate the wire length(s) by multiplying it with a small factor. Update matrices A and B and the objective function. If any entry in matrix B becomes negative, go to step 4.
3. Solve the linear programming problem stated by (10), (11) and (12). If the solver finds an optimal solution for w_{ab}, w_{bc}, w_{cd} and w_{ad} , calculate $w_{as} \sim w_{ds}$ using (9), connect the tapping point to the clock driver and exit; else go to step 2.
4. Relax the target delay D and go back to step 2.

3. Fast analysis and heuristic optimization

The mesh/tree clock network constructed using the above algorithm assumes minimum wire width for the underlying zero-skew trees. We realize that the transition time, defined by the delays between 10% of Vdd and 90% of Vdd, is another important factor that affects the performance of the clock network. It determines the maximum clock frequency of the chip. In our work, this is achieved by simultaneous buffer insertion and wire widening optimization. A heuristic optimization procedure that increases the wire width with the largest transition time reduction while maintaining the worst-case skew within a small constraint is developed. This procedure is similar to the TILOS heuristic [7] for transistor sizing. Due to the well-known limitations of the Elmore delay model, we use a core timing calculator based on a higher order model. The transition time and skew optimization is especially crucial for deep sub-micron networks to work reliably. For the mesh/tree model, the analysis can be performed using a hierarchical model reduction and simulation method similar to the procedure in [16].

To calculate the delay sensitivities and the skew sensitivities, the conventional adjoint sensitivity analysis is first applied to calculate the partial derivatives of voltages with respect to all the wire widths in the network. Using the formula described in [4], the delay sensitivity of sink i with respect to wire width w_{pq} can be computed as

$$\frac{\partial D_i}{\partial w_{pq}} = -\frac{\partial V_i / \partial w_{pq}}{\partial V_i / \partial t} \Big|_{t=D_i} \quad (14)$$

The same method can be used to compute $\partial D_j / \partial w_{pq}$. From these, we obtain the skew sensitivity as well as the transition time sensitivity.

In each iteration of the heuristic optimization, if the worst-case skew between sink i and j is greater than the skew constraint, the wire width with the most negative $\frac{\partial D_i}{\partial w}$, positive $\frac{\partial D_j}{\partial w}$ or negative $\frac{\partial D_j}{\partial w}$ with its absolute value smaller than $\frac{\partial D_i}{\partial w}$ is bumped up with a small factor (<1.1). Otherwise, the wire width with the most negative $\frac{\partial SlewRate}{\partial w}$ is bumped up.

If no negative skew or transition time sensitivity exists at certain iteration, increasing any of the wire widths will increase the skew or the transition time, which means this network cannot be optimized any further by wire widening only.

4. Experimental results

We have applied our clock network design scheme to the five benchmark examples in [17] under the 0.18micron technology. For the experiments, we scaled the wire widths while maintaining the same sink load capacitances under the assumption that under scaling theory, for a fixed area, the reduction in the feature size cancels the increase of the number of capacitive loads. Both the mesh/tree construction algorithm and heuristic simultaneous buffer insertion and wire width optimization procedure have been implemented using C++. All of the experiments are performed on Sun Ultra-60 workstations.

Table I Comparison between structures of clock nets

	# of Pin	Prop Delay (ns)	Skew (ps)	Tran Time (ns)	Area (cm ²)	Num of buf	CPU time (sec)
Hybrid structure							
R1	267	0.105	3.194	0.221	2.29e-4	36	5.05
R2	598	0.127	2.256	0.243	3.14e-4	80	11.85
R3	862	0.164	2.054	0.337	4.34e-4	82	21.04
R4	1903	0.174	2.616	0.333	4.27e-4	220	39.50
R5	3101	0.209	1.741	0.389	4.67e-4	274	71.57
Tree structure							
R1	267	0.115	6.676	0.261	1.44e-4	36	5.18
R2	598	0.140	1.912	0.289	1.63e-4	80	13.51
R3	862	0.175	2.462	0.397	1.62e-4	82	20.88
R4	1903	0.215	2.772	0.475	2.34e-4	220	39.98
R5	3101	0.230	3.098	0.469	2.46e-4	274	72.20

We use Tsay's [17] zero-skew algorithm to construct the underlying buffered trees. Buffers are inserted while merging the two sub-trees. This creates a buffered binary tree for each case. A one-level mesh constructed using our linear programming based algorithm then connects the four underlying trees. We also constructed the full tree structure with the same total number of buffers for comparison. Table I shows the performance of our zero-skew mesh/tree structures and the comparison with the full tree results using Tsay's method. The propagation delay and transition time have been reduced by an average of 10% and 15% respectively. Under the Elmore delay metric, the skew of each example is zero, by construction. We further observed the actual delays and skews using the HPRIMA simulator and these are listed in the "Prop Delay" and "Skew" columns in Table I. The transition time is listed in the "Tran time" column. The CPU

times are shown in the column of “CPU Time” and it is seen to be fast for all these examples. The total wire area for the top-most structure is listed.

Table II Wire width optimization results

	Prop Delay (ns)	Skew (ps)	Tran Time (ns)	Area (cm ²)	Num of Itr	CPU Time (sec)
R1	0.097	1.253	0.197	2.59e-4	28	1.25
R2	0.122	1.682	0.220	4.74e-4	87	3.83
R3	0.163	1.641	0.328	4.73e-4	21	1.70
R4	0.167	2.890	0.296	6.66e-4	82	8.72
R5	0.196	1.935	0.338	5.79e-4	41	5.37

Table II shows the wire width optimization results. Total number of iteration is listed in the “Num of Itr” column. It can be seen that after optimization, the propagation delay and the transition time of all the networks are both efficiently reduced while the maximum skew is within $3ps$.

Table III Process variation results

	Nom Skew (hybrid) (ps)	Nom Skew (tree) (ps)	Skew Var Mean (hybrid) (ps)	Skew Var Mean (tree) (ps)	Std (hybrid) (ps)	Std (tree) (ps)
R1	3.194	6.676	0.238	1.949	0.026	0.071
R2	2.256	1.912	0.220	1.143	0.025	0.098
R3	2.054	2.462	0.238	0.852	0.029	0.058
R4	2.616	2.772	0.229	0.477	0.050	0.110
R5	1.741	3.098	0.670	1.370	0.030	0.107

Table III shows the statistic results for process variations in the top-most level of structures (hybrid and pure tree). Both spatial and random wire width variations are applied to each wire. The nominal skew, which is the skew without any wire width variation is listed in the “Nom Skew” columns. The mean of variations away from the nominal skew is shown in the “Skew Var Mean” columns. In all these examples, both the skew variation and the standard deviation of skews of the hybrid structure are less than the pure tree structure. It can be seen that the hybrid structure is less sensitive to process variation than the pure tree structure and it works more reliably.

Table IV Comparison with the IBM structure

	# of pins	Structure	Hybrid structure			
			Prop Delay (ns)	Skew (ps)	Tran Time (ns)	Area (cm ²)
R6	64	Hybrid	0.191	0.080	0.536	9.50e-4
R6	64	IBM	0.348	2.550	1.068	9.53e-4
R7	64	Hybrid	0.377	0.260	1.128	8.22e-4
R7	64	IBM	1.059	4.500	3.317	8.20e-4

Table IV shows the comparison of our hybrid structure with the IBM structure discussed in [14]. A total of 64(8x8) uniformly distributed sinks is generated, with the load capacitance ranging from 31fF to 67fF in R6 and from 35fF to 430fF in R7. Connecting all the leaf nodes of the pure tree structure into a full 8x8 mesh generates the IBM structure. The upper level tree is optimized such that both structures have almost the same total wire area. The same clock driver is applied to both structures and HSPICE simulation is performed. From

the two small examples, we can see that the hybrid structure shows better performance than the IBM structure.

5. Conclusion

This paper has proposed a versatile procedure for building hybrid mesh/tree clock networks. The theoretical foundation laid down in this work may lead to further developments that make it easier to build mesh-based zero-skew clock structures.

6. Acknowledgements

The authors would like to thank Sani R. Nassif of IBM and Duncan M. Walker of Texas A&M University for help in setting up the process variation experiments.

References

- [1] K. Bernstein, *et al.*, *High Speed CMOS Design Styles*, Kluwer Academic Publishers, Norwell, Massachusetts, 1998.
- [2] T. Chao, *et al.*, “Zero Skew Clock Routing With Minimum Wirelength”, *IEEE Transactions on Circuits and Systems*, vol. 39, pp. 799-814, Nov. 1992.
- [3] L. O. Chua and P. M. Lin, *Computer-Aided Analysis of Electronic Circuits*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.
- [4] A. R. Conn, *et al.*, “Circuit Optimization via Adjoint Lagrangians,” *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 281 – 288, 1997.
- [5] M. P. Desai, R. Cvijetic, and J. Jensen, “Sizing of Clock Distribution Networks for High Performance CPU Chips,” *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 389-394, 1996.
- [6] J. P. Fishburn, “Clock skew optimization,” *IEEE Transactions on Computers*, vol. 39, pp. 945-951, Jul. 1990.
- [7] J. P. Fishburn and A. E. Dunlop, “TILOS: A posynomial programming approach to transistor sizing,” *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 326-328, 1985.
- [8] S. Lin and C. K. Wong, “Process-Variation-Tolerant Clock Skew Minimization,” *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 284-288, 1994.
- [9] N. Menezes, *et al.*, “Skew Reduction in Clock Trees Using Wire Width Optimization,” *IEEE Custom Integrated Circuits Conference*, pp. 9.6.1-9.6.4, 1993.
- [10] A. Odabasioglu, M. Celik, and L. T. Pileggi, “PRIMA: Passive reduced-order interconnect macromodeling algorithm,” *IEEE Transactions on Computer-Aided Design*, vol. 17, pp. 645-654, Aug. 1998.
- [11] L. T. Pileggi and R. Rohrer, “Asymptotic Waveform Evaluation for timing analysis,” *IEEE Transactions on Computer-Aided Design*, pp. 352 - 366, April, 1990
- [12] S. Pulella, *et al.*, “Skew and Delay Optimization for Reliable Buffered Clock Trees,” *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 556-562, 1993.
- [13] S. Pulella, N. Menezes and L. T. Pileggi, “Reliable Non-zero Skew Clock Trees Using Wire Width Optimization,” *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 165-170, 1993.
- [14] P. J. Restle, A. E. Ruehli and S. G. Walker, “Multi-GHz interconnect effects in microprocessors,” *Proceedings of the ACM International Symposium on Physical Design*, pp. 93-97, 2001.
- [15] J. C. Shah, *et al.*, “An algorithm for simulating power/ground networks using Padé approximants and its symbolic implementation,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, pp. 1372-1382, Oct. 1998.
- [16] H. Su, K. H. Gala and S. S. Sapatnekar, “Fast Analysis and Optimization of Power/Ground Networks,” *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2000.
- [17] R. Tsay, “An Exact Zero-Skew Clock Routing Algorithm,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 242-249, Feb. 1993.