JCSE Journal of Computing Science and Engineering

Regular Paper
Journal of Computing Science and Engineering,
Vol. 6, No. 4, December 2012, pp. 257-266

Design Methodologies for Reliable Clock Networks

Deokjin Joo, Minseok Kang, and Taewhan Kim*
Department of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea
jdj@ssl.snu.ac.kr, ikado@ssl.snu.ac.kr, tkim@ssl.snu.ac.kr

Abstract
This paper overviews clock design problems related to the circuit reliability in deep submicron design technology. The topics include the clock polarity assignment problem for reducing peak power/ground noise, clock mesh network design problem for tolerating clock delay variation, electromagnetic interference aware clock optimization problem, adjustable delay buffer allocation and assignment problem to support multiple voltage mode designs, and the state encoding problem for reducing peak current in sequential elements. The last topic belongs to finite state machine (FSM) design and is not directly related to the clock design, but it can be viewed that reducing noise at the sequential elements driven by clock signal is contained in the spectrum of reliable circuit design from the clock source down to sequential elements.

Category: Embedded computing

Keywords: Clock design; System reliability; System-on-chip; Embedded systems

I. INTRODUCTION

In a synchronous digital system, the clock signal is used to define a time reference for the movement of data in the system. The clock distribution network distributes the clock signal(s) from a clock source to all sequential elements which require it. Thus, the clock function is vital to the operation of the synchronous system. Nowadays, much more attention has been paid to clock related design than ever before. This is because as the clock frequency increases over a 1 GHz with low supply voltage, a small noise on the clock signal causes a transient function error or even a drastic system failure (the noise comes from many factors such as current charge/discharge variation and temperature variation). One of the most important clock design issues is analyzing the clock signals' behavior, and mitigating the adverse impact of clock noise on circuit reliability.

This work overviews a number of important clock network optimization problems and the proposed techniques with regard to the circuit reliability in deep submicron design technology. The following subsections cover 1) the clock polarity assignment problem for reducing peak current noise on the clock tree, 2) the clock mesh network design problem for tolerating the clock skew variation, 3) the electromagnetic interference (EMI) aware clock optimization problem, 4) the adjustable delay buffer (ADB) allocation and assignment problem that is useful in the multiple voltage modes design environment and finally 5) the finite state machine (FSM) state encoding problem for reducing peak current in the sequential elements in FSM. Note that the last topic is not directly related to the clock design issue, but it can be regarded that reducing noise at the activation of sequential elements driven by a clock signal is contained within the spectrum of achieving reliable circuits from the clock source down to sequential elements.

Open Access http://dx.doi.org/10.5626/JCSE.2012.6.4.257 http://jcse.kiise.org

Received 30 October 2012, **Accepted** 16 November 2012

*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

II. CLOCK TREE SYNTHESIS OVERVIEW

A typical clock tree synthesis (CTS) process consists of three phases: abstract tree topology generation, clock tree routing, and buffer insertion.

Fig. 1 shows the flow of the methods of means and medians (MMM) algorithm [1]. It takes the clock sink locations as input and generates an abstract clock tree topology by recursively partitioning the sinks into two subregions until there are at most two sinks on each subregion. The partitions are performed either in the x direction or y direction, dividing the number of sinks equally using the concept of median. The partitions are then merged back in reverse order, forming an abstract tree topology where the merging points are determined using the concept of mean (i.e., center of mass) of the sink locations on the subregions to be merged. The more accurate determination of the location of merging points (i.e., the internal nodes of tree) is performed in the wire routing step.

With the abstract topology obtained, the actual wire routing from the clock source toward the sinks is accomplished by other algorithms. One of the most notable algorithms is the deferred merge embedding (DME) algorithm [2, 3], which ensures zero clock skew while minimizing the total clock wire length (the clock skew refers to the delay difference between the earliest arrival time and latest arrival time to sinks). The DME algorithm runs in two stages; one is a bottom-up process from sinks and the next is a top- down process from the root of clock tree.

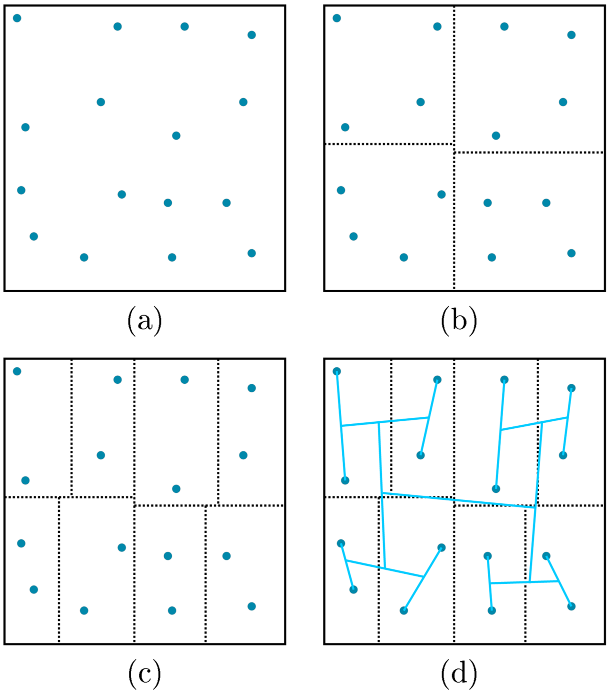


Fig. 1. The flow of the methods of means and medians algorithm [1]. (a) The location of all input clock sinks is fed as input. (b, c) Sinks are equally, using the concept of median of sink locations, partitioned into subregions (arbitrarily in the x- or y-direction) until at most two sinks are left on each subregion. (d) The partitions are merged back in reverse order, using the concept of mean (or center of mass) of the sink locations on the two subregions to be merged to produce an abstract tree topology.

Fig. 2a and b show the bottom-up stage in which the algorithm builds *merging segments* or *merging regions*, if a bounded clock skew constraint is employed. Each point in the merging segment or region satisfies the clock skew constraint for the clock subtree rooted at the point, thus the points in the merging segment or region are the candidates for the wire branching location of the corresponding internal node in the abstract tree topology. Fig. 2c and d illustrate the top-down stage. The top-down stage finalizes the exact positions of branching locations. While all the points in merging segments or regions satisfy the clock skew constraint, depending on the point selection, the wire length may change and it is the top-down stage that ensures the minimal length of total wires.

The final step of CTS is to insert buffers into the clock tree. The clock slew-rate, which is the sharpness of the clock signal voltage rise and fall, must be controlled in modern chip design. With large clock trees, the wire capacitance is too large to be driven by one large buffer placed at the root of the clock tree. Hence, buffers need to be inserted into the clock tree to meet both the slew and skew constraints. Many works have combined the first

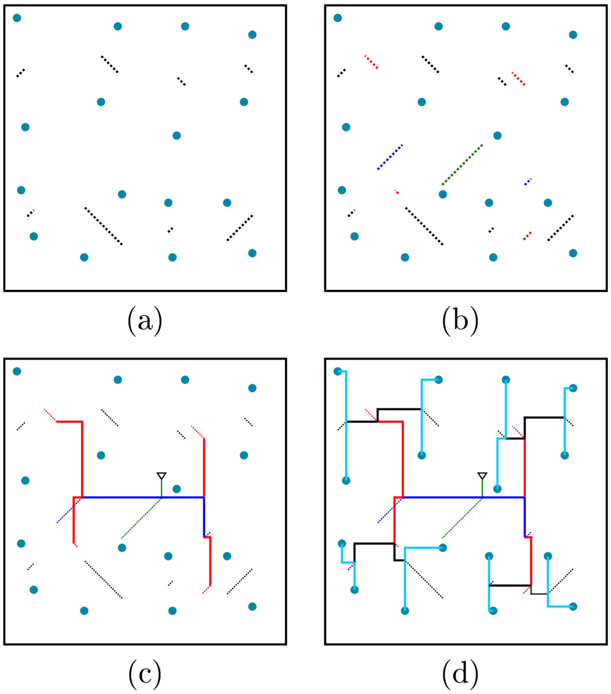


Fig. 2. The process of the deferred merge embedding algorithm. (a, b) Merging segments, which are the candidates of points of clock tree branching locations, are constructed in a bottom-up manner. (c, d) Exact branching locations are identified and wire length minimization is performed in a top-down manner.

two phases with the buffer insertion [4-6].

Recently, some works attempted to integrate circuit simulation into the CTS process while trading off between the accuracy of the delay model and the quality of CTS [7]. On the other hand, instead of sacrificing simulation time with CTS quality, some approaches have proposed the adaptation of the independent delay model. This is accomplished by carefully constructing clock trees to have a symmetrical structure that guarantees zero clock skew throughout the clock tree. Shih and Chang [8] met near the zero skew (<2.1 ps) goal while achieving orders of magnitude runtime speedup.

III. RELIABILITY AWARE CLOCK DESIGN TECHNIQUES

A. Clock Polarity Assignment

In synchronous circuits, clock trees and its clocked loads are the major sources of on-chip noise (power/ground voltage fluctuations) since they switch simultaneously near the rising and/or falling edge of the clock signal and draw a significant amount of current from the power/ground rail. To accommodate this simultaneous switching, Vittal et al. [9] and Vuillod et al. [10] took advantage of clock skew to scatter the clock signal arrival times along the time domain.

Thereafter, Nieh et al. [11] introduced additional flexibility to this scattering, by firstly proposing the polarity assignment technique. They demonstrated that mixing buffers and inverters for clock buffering elements can disperse noise over rising and falling edges of the clock as shown in Fig. 3. The approach assigned half of the buffering elements to a negative polarity and the other half positive by replacing one of the two buffers that are connected to the clock source with an inverter. However, although the total peak current is significantly reduced, since power/ground noise is a local effect, the problem remained largely unsolved. Samanta et al. [12] mixed buffers and inverters throughout the clock tree structure so that for each local region about half of the buffering elements have positive polarity and the other half negative polarity. This approach, however, while greatly reducing noise, is likely to introduce large clock skew. According to more recent data [13], mixing buffers and inverters for non-leaf elements resulted in an average clock skew of 592 ps. Chen et al. [14] observed that noise from leaf buffering elements dominates that from non-leaf buffering elements. This phenomenon is more prominent when the clock tree is not a binary tree where one non-leaf buffering element has more than two leaf buffering elements attached. Hence, they proposed to assign polarity to leaf buffering elements only and assigned half of the leaf nodes to negative polarity, with the objective of minimizing clock skew. However, this approach only heuristically solved the noise reduction problem in that it does not consider the buffer load which affects the peak value of the noise current. Jang et al. [15] proposed and solved a combined problem of buffer sizing and polarity assignment with the proof of the nondeterministic polynomial time (NP)-completeness of the polarity assignment problem. It retrieves all assignment combinations under the clock skew constraint and selects the one with the lowest noise. Very recently, Joo and Kim [16] proposed a fine-grained polarity assignment to overcome the two limitations of the prior works, which are the unawareness of the signal delay (i.e., arrival time) differences to the leaf buffering elements and the ignorance of the effect of the current fluctuation of non-leaf buffering elements on the total peak current waveform.

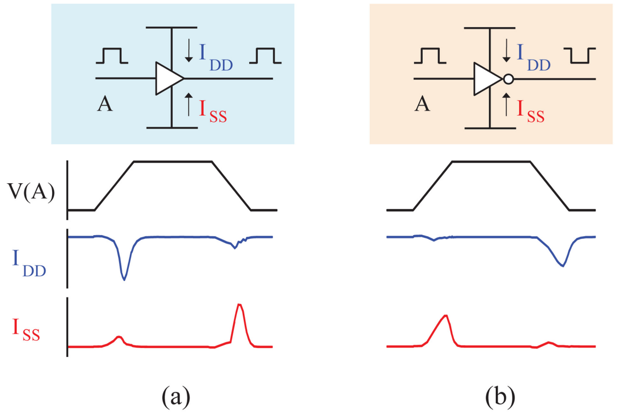
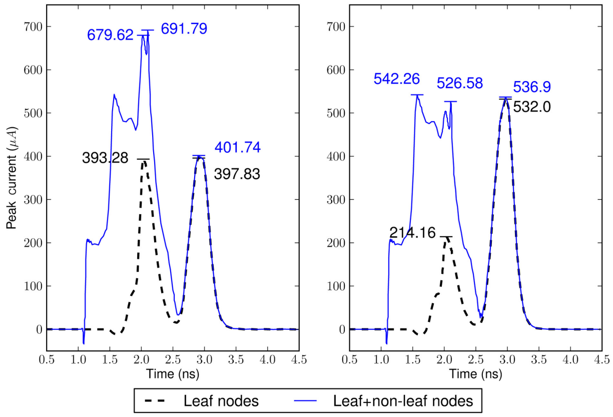


Fig. 3. Current profiles for a buffer and an inverter. (a) Buffers draw larger current from the power rail at the rising edge of the clock and discharges the current to the ground rail at the falling edge. (b) For the inverter, the opposite phenomenon occurs.

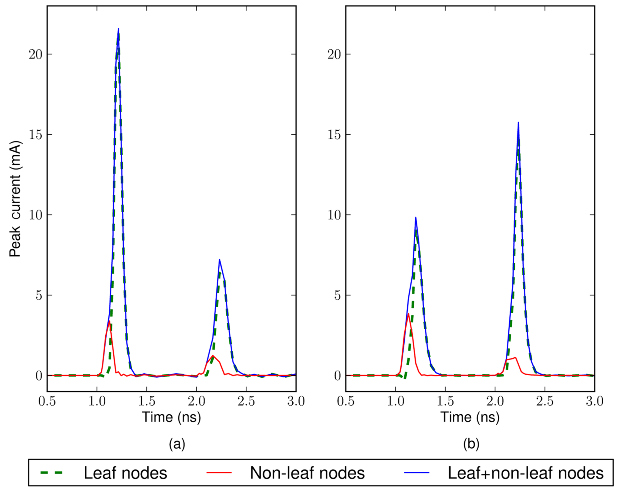
Fig. 4 illustrates the difference between non-leaf blind and non-leaf aware polarity assignments. In Fig. 4a, the black dotted line which is the current waveform from leaf nodes is well balanced, indicating that the noise is minimized for the leaf nodes. However, the solid line with blue color which is the total noise waveform of both the leaf nodes and the non-leaf nodes is not balanced. The combined noise causes a higher peak current value than the original expectation that comes from the consideration of leaf nodes only. Fig. 4b shows an improved result in which the non-leaf noise is considered. At leaf nodes, more current noise is migrated to the falling edge of the clock signal, yielding a greater margin for the non-leaf node noise. The results of Fig. 4 suggest that polarity assignment algorithms need to consider the effect of the noise caused by non-leaf nodes on the leaf nodes, even when only assigning polarity to the leaf nodes only.

Fig. 5 compares the polarity assignment results by Jang and Kim's algorithm [15] and Joo and Kim's fine-grained approach [16]. Joo and Kim's modeling captured more details of the noise waveform, including smaller bumps at the opposite clock edge of the larger peak noise,



(a) (b)

Fig. 4. (a) Current waveforms by non-leaf nodes' noise unaware optimal polarity assignment to leaf nodes. (b) Current waveforms resulting from non-leaf nodes' noise *aware* optimal polarity assignment to leaf nodes. Dark dotted line is the current waveform from leaf nodes only while blue solid line shows the total current from all clock nodes.



(a)

(b)

Fig. 5. Current waveforms for design s35932. (a) Result of polarity assignment without considering the noise effect by non-leaf nodes in Jang and Kim [15]. (b) Result of polarity assignment with the consideration of the noise effect by non-leaf nodes in Joo and Kim [16].

arrival time differences, and non-leaf element noise. Thus, more of the buffering elements were assigned as inverters, showing better balanced noise levels at the rising and falling edge of the clock.

In addition, there are a number of works which have considered other design factors to improve the quality of polarity assignment. Kang and Kim [17] considered a statistical delay variation to increase yield during polarity assignment. Ryu and Kim [18] took a different approach by assigning polarity to leaf nodes first and then routing the clock tree. Lu and Taskin [19] exploited exclusive-or (XOR) gates, instead of inverters and buffers, so that polarity may be controlled at runtime.

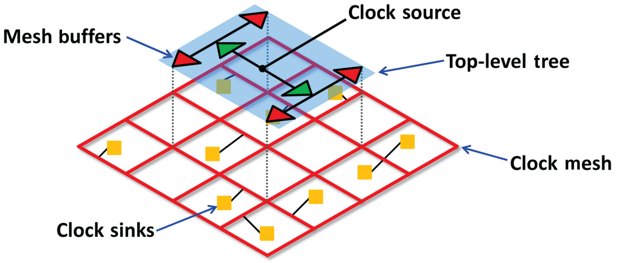


Fig. 6. Mesh-based clock distribution network.

B. Clock Mesh Network Design

As complementary metal-oxide-semiconductor (CMOS) process scaling continues under deep submicron technology while clock frequency increases, the variation effect on clock skew becomes significant. It is known that the delay variation on the interconnect can cause up to 25% variation in clock skew [20], implying the necessity of controlling the variation effect during the clock network design. The mesh-based clock distribution network, as depicted in Fig. 6, is one of the solutions that can effectively mitigate the clock skew variation since multiple paths from the clock source to a sink are able to compensate the different clock arrival times.

However, the multiple paths are likely to generate short circuit currents between mesh buffers which have different clock arrival times. Moreover, the mesh-based clock network requires much more wires than the tree-based clock network in order to support multiple clock signal paths. Consequently, most of related works have focused on reducing the amount of power consumption and wire resource while maintaining clock skew variation tolerance. Venkataraman et al. [21] decided the location and size of buffers first, and then reduced mesh wires; they solved the mesh buffer placement and sizing problem by formulating it into a weighted set-cover problem. Precisely, for each node (i.e., buffer) in the clock mesh they defined a so called covering region, which refers to the set of nodes that the node can drive. Subsequently, they locate a minimal set of covering regions that can cover all nodes in the clock mesh. They employed a greedy algorithm which iteratively picks a covering region that covers the largest number of uncovered nodes. The mesh reduction problem was then solved by formulating it into the Steiner network problem [22]; they iteratively removed wire segments while maintaining a certain level of redundancy, i.e., every clock sink should have at least k closest clock buffers within a distance of L_{max}. After mesh reduction is performed, readjustment of the buffer size is attempted as a post-processing. On the other hand, Rajaram and Pan [23] suggested an initial mesh planning; they expressed the total wirelength and worst-case clock skew as functions of mesh size, by which they created an initial mesh size under the worst-case clock skew constraint. In addition, they improved the cost function used in the greedy algorithm in [21] by considering a potential effect of buffer insertion on mesh and low-pass filter characteristics of an RC mesh. They also proposed a network sensitivity-based mesh reduction algorithm in which the delay sensitivity of each sink is calculated in terms of the width of the wire segment. By using this delay sensitivity, they removed the wire segments that cause little effect on clock skew.

To demonstrate the variation tolerance of the clock mesh, we compare the clock mesh with the clock tree in terms of the clock skew and resource usage. We implemented the clock mesh with a given mesh size and inserted buffers by using the algorithm of buffer placement and sizing in [21]. Then, the clock trees are generated by applying the algorithm in [3, 6]. We tested clock meshes and clock trees on ISCAS89 benchmark circuits. We used 12 buffer sizes with maximum-capacitance ranging from 60 to 500 fF under a slew constraint of 100 ps. The technology parameters and transistor models were based on 45 nm predictive technology [24]. We consider 4 variation parameters: power supply voltage, buffer channel length, wire width, and sink capacitance. All variation parameters are varied with 5% standard deviation around its nominal value. We also consider the spatial correlation by the principal component analysis [25]. The columns in Table 1 are benchmark circuit name, given mesh size ('Tree' for the result of clock tree), mean and standard deviation of the clock skew (measured by HSPICE Monte-Carlo simulations), total wirelength (WL), buffering area (BA), and power consumption (PWR). The last two rows in Table 1 are the average ratio of clock trees and clock meshes. Table 1 indicates that the clock mesh can reduce 50% of worst-case clock skew ($\mu + 3\sigma$) on average with the increase of 42% of total WL, 29% of BA, and 37% of PWR.

Table 1. Comparison of clock tree and clock mesh in terms of clock skew, wire length (WL), buffer area (BA), power consumption (PWL)

Cir. (#sinks)	Size	μ_{sk}/σ_{sk} (ps)	WL (μm)	BA (μm^2)	PWR (mW)
s5278	Tree	25.3/12.8	28360	112	6.94
(180)	5X5	4.8/2.1	41342	152	9.61
s9234	Tree	25.0/8.5	40583	153	9.71
(212)	6X6	6.8/2.8	56974	212	13.50
s13207	Tree	40.7/17.0	96871	396	23.62
(639)	10X10	18.7/10.7	140816	493	32.94
s15850	Tree	39.6/11.6	111555	434	26.78
(535)	10X10	19.4/9.7	158921	547	37.24
s35932	Tree	100.4/37.3	322534	1276	77.54
(1729)	18X18	66.3/26.5	441964	1602	104.07
s38417	Tree	87.3/30.7	269467	1082	65.69
(1637)	18X18	49.4/22.3	373647	1366	88.06
Avg.	Tree	1.00/1.00	1.00	1.00	1.00
Ratio	Mesh	**0.44/0.57**	**1.42**	**1.29**	**1.37**

C. Electromagnetic Inference Aware Clock Design

Increasing the requirement for high performance circuits and the enforcement of strict governmental regulation leads to a new design solution for EMI. Electromagnetic radiation from an electronic circuit can be categorized into two forms [26]: *differential-mode radiation* and *common-mode radiation*. Differential mode radiation occurs in the current loop, which is related to on-chip wiring, the package and enclosure, and board wiring among other things. In contrast, common-mode radiation is the result of voltage level difference. Thus, common-mode radiation is closely related to the signals on data and clock wires.

EMI-aware clock optimization is a clock network synthesis method with the objective of reducing EM emission by controlling clock parameters. The key clock parameters are slew rate and clock skew. Slew rate is the rise/fall time in the clock signal which typically has a trapezoidal waveform. With Spectral analysis, the clock signal can be represented in the sum of series regarding sine and cosine functions. The amplitude of high frequency functions critically depends on the speed of slew rate [27]. In other words, a clock signal with a fast slew rate has larger amplitude in high frequencies compared to those with a slower slew rate. Previous works demonstrate that in order to reduce total EMI it is very effective to decrease the slew rate of the clock signal [27-29]. Pandini et al. [27] proposed a method to decrease the slew rate by manually removing buffers and inverters with higher driving strength from the target library. Hu and Guthaus [29] proposed a method of weakening buffer driving strength by setting the constraint of minimum slew rate during clock tree synthesis. They proposed an incremental dynamic programming algorithm to determine buffer sizing and positioning for a clock tree, considering metrics such as skew and power used in [27]. EM emission is also primarily generated by simultaneous switching noise (SSN). In the clock network, sink buffers and connected flip-flops toggle almost at the same time, creating a large current pulse on the power network, and fast current variation increases di/dt, causing a high SSN. Note that peak current can be controlled by spreading the activity of buffers and flip-flops within a time period, but it causes a clock skew variation. Vuillod et al. [10] proposed a solution to reduce peak current by relaxing the clock skew constraint. Pandini et al. [27] performed a theoretical analysis on power/ground noise by describing noise using a periodic triangular pulse, and demonstrated that increasing skew bound is effective in reducing EM emission.

D. Adjustable Delay Buffer Allocation

Traditional clock optimization techniques are mainly based on the environment in which all elements in a circuit use one fixed operating voltage. However, a recent technological trend requires multiple supply voltages to allow the voltage level applied to a circuit to be dynamically changed. In this multi-voltage mode design, some part of the circuit could operate at high voltage when the associated module is required to complete its processing quickly and at low voltage when timing requirement can be relaxed and reducing power is more important. When the supplied voltage switches from high to low (or low to high), the delay of all logic elements including buffers in the clock network on the part of the chip also varies. One serious problem in this multi-voltage mode design is the clock skew variation on the clock network.

To tackle the clock skew variation problem, Su et al. [30] proposed a methodology of dealing with clock skew optimization in which they proposed using ADB, which is a specially designed buffer so that its delay can be controlled dynamically. The general structure of ADB is shown in Fig. 7. It is composed of a normal buffer, internal capacitor bank, and a capacitor bank controller. The delay of ADB could be adjusted by activating the internal capacitor bank. That is, the adjusted delay of ADB is determined by the amount of activated capacitors in the capacitor bank that is controlled by the capacitor bank controller and its control input.

Fig. 8 demonstrates how ADBs can be used to fix clock skew violations. In Fig. 8a, an initial clock tree with clock signal arrival times for each clock sink in two power modes is given. With the clock skew bound of 10, there is one clock skew violation between FF_1 and FF_6 in power mode 1. By replacing buffer B_1 with an ADB and inserting an additional delay of 3, this violation can be fixed. For power mode 2, there are two violations, one between FF_3 and FF_6 and the other between FF_2 and FF_6. By allocating ADBs at B_1 and B_3 and assigning their delay values with 4 and 1, respectively, the clock skew violation is removed. The resulting ADB allocation is shown in Fig. 8b.

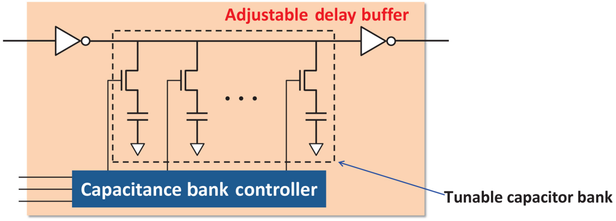
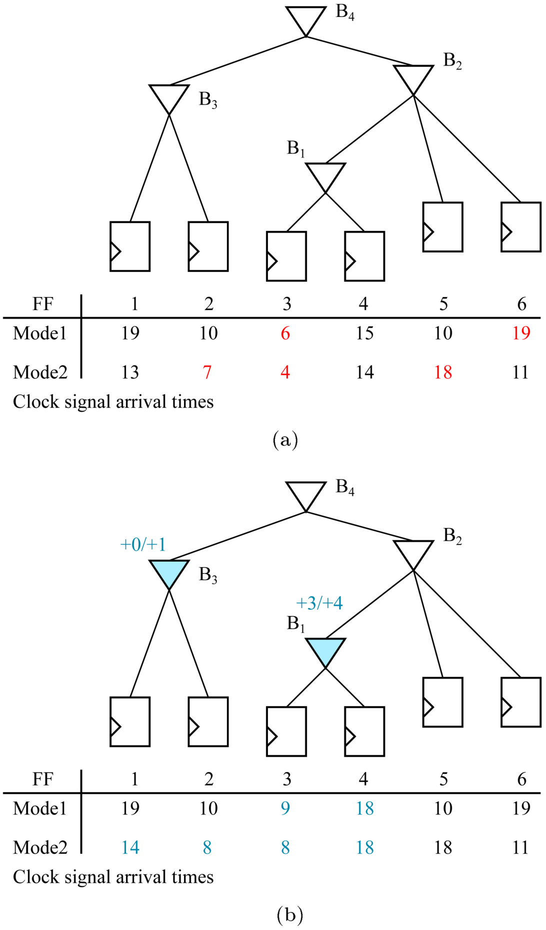


Fig. 7. Structure of adjustable delay buffer [31].



FF	1	2	3	4	5	6
Mode1	19	10	6	15	10	19
Mode2	13	7	4	14	18	11

Clock signal arrival times

FF	1	2	3	4	5	6
Mode1	19	10	9	18	10	19
Mode2	14	8	8	18	18	11

Clock signal arrival times

Fig. 8. An example clock tree with two power modes and the information of clock signal arrival times at clock sinks. (a) Under clock skew bound of 10, there exist clock skew violations. (b) Two adjustable delay buffers are allocated at B_1 and B_3 to resolve the violations.

The problem to be solved in using ADBs in a multi-voltage mode design is to minimize the cost of ADBs to be used since an ADB has more transistors than a normal buffer due to the internal capacitor bank and controller. The approach in [30] focused on the power mode with the worst violation of the clock skew constraint and resolved the worst clock skew by adding ADBs in a greedy manner. The approach was then repeated until there was no clock skew violation for every power mode. While the approach can efficiently find an ADB allocation with no clock skew violation, an optimal ADB allocation is not guaranteed due to the inherent limitation induced by the iterative heuristic. Later on, Lim and Kim [32] improved the work in [30] by proposing a linear time optimal algorithm for the ADB allocation and delay value assignment. Initially, they replaced all buffers in the clock tree with ADBs and then removed unnecessary ADBs through a comprehensive analysis of the timing relation between ADBs. They performed the analysis in a bottom-up fashion in the clock tree and showed that the ADB allocation is a polynomial-time optimal algorithm, bounded by $O(N \cdot K)$ where N and K are the number of buffers in the clock tree and the number of power modes, respectively.

Recently, Lim et al. [32] addressed several practical design issues related to the non-ideal characteristics of ADBs. First, non-ideal ADBs have non-uniform discrete delay steps which harm the optimality of ADB allocation in [32]. Second, ADBs have a non-zero base delay that is larger than that of typical buffers even when all the capacitors in the bank are deactivated. Third, the output slew increases as more of the capacitors in the bank become active. Last, further ADB area reduction is possible without affecting the number of ADBs currently allocated. They addressed these issues by proposing heuristic iterative refinement algorithms and proposing a new modified design of the ADB circuit.

E. Noise Aware FSM State Encoding

Since FSM is a synchronous component controlled by the clock signal of a circuit, it also draws the peak current on the power line (V_{DD}) and ground line (V_{ss}) of the circuit when state transition occurs. Thus, it is also important to reduce the number of FSM flip-flops in the state register that switch simultaneously. Peak current in state encoding can be explained using the example given in Fig. 9, which shows state transition graphs (STGs) of a FSM. The maximum switching on the left FSM (STG$_1$) occurs when state changes from S_0 to S_4 (three 0-to-1 transitions). Meanwhile, the maximum switching on the right FSM (STG$_2$) occurs when state changes from S_3 to S_0 (two 1-to-0 transitions). This example demonstrates that proper state encoding may reduce the circuit noise.

Huang et al. [33] start from the encoding result produced by the scheme in [34] which targets reducing the total power (i.e., total switching activity) of an input STG. The idea of reducing peak switching among state transitions is to decide whether the value of each bit position in the previously encoded registers is to be complemented or not. For example, if the solution selects to complement bit positions 0 and 3, then all the values in bit positions 0 and 3 in state encoding are complemented. They formulated the problem of determining bit complementing/uncomplementing into an integer linear programming (ILP), targeted towards minimizing the larger of the maximum numbers of bit positions that switch from 0 to 1 among state transitions and the maximum number of bit positions that switch from 1 to 0.

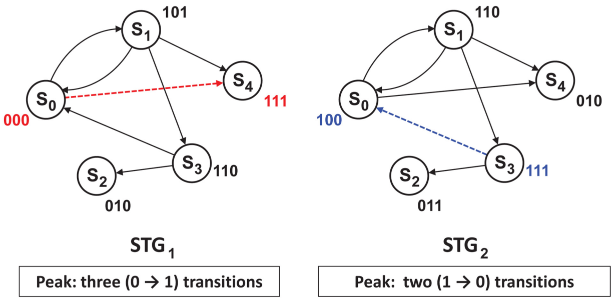

S1
101
S4
111
S0
000
S3
110
S2
010
STG 1
Peak: three (0 → 1) transitions
S1
110
S4
010
S0
100
S3
111
S2
011
STG 2
Peak: two (1 → 0) transitions

Fig. 9. Two state transition graphs (STGs) of finite state machine.

For n-state FSM which has an m-bit code length, the approach explores design space as large as 2^m. Lee and Kim [35] improved Huang's work by considering the trade-off between power and noise, in which they minimized peak switching, followed by minimizing total switching (i.e., minimizing total dynamic power consumption). They formulated the peak switching minimization problem into a satisfiability (SAT) problem with pseudo-boolean (PB) expressions, and solved it by using a PB-Solver [36]. Then, they iteratively updated the PB expression to extract a state transition which results in the highest switching, and running PB-Solver with the additional conjunctive normal form (CNF) constraints. On the other hand, Gu et al. [37], similar to the work in [33], started from an encoded FSM with minimal total switching, but they identified a set of transitions (called working set S) that cause peak current and applied both state-replication and state-encoding to S. State-replication replicates a state to assign another code to each replicated state with the proper generation of state transitions while state-encoding re-encodes the given code of a state to another unused code. With an input FSM with minimum total switching, by applying state-replication and state-encoding in a combined manner iteratively, they reduced the peak switching.

IV. CONCLUSION

As the process technology scales down, the variation or sensitivity of noise and delay in the clock network becomes worse and worse, which in fact causes a drastic impact on the reliability of the circuit. This paper reviewed several important clock related design problems and the existing techniques that are essential to the highly reliable circuit design. In addition to the introduced design optimization and synthesis issues, there exist other issues that are also important to tolerate or mitigate circuit noise. Those examples are 3D clock design considering through silicon via (TSV) design variation and circuit reliability issues under the power-gated or clock-gated design environment. We hope that this survey helps to find or assess practically useful solutions to the important clock optimization problems that arise in the designs supporting diverse platforms, applications or environments (e.g., [38, 39]).

ACKNOWLEDGMENTS

This work was supported in part by the Basic Science Research Program through the National Research Foundation (Grant 2011-0029805), and by the Center for Integrated Smart Sensors funded by the Ministry of Education, Science, and Technology as the Global Frontier Project (CISS 2011-0031863) and supported by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center Support Program supervised by the National IT Industry Promotion Agency (Grant NIPA-2012-H0301-12-1011).

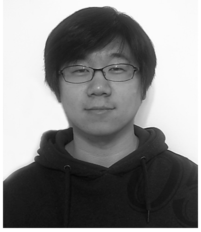
REFERENCES

1. M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock routing for high-performance ICs," *Proceedings of the 27th ACM/IEEE Design Automation Conference*, Orlando, FL, 1990, pp. 573-579.
2. T. H. Chao, Y. C. Hsu, J. M. Ho, and A. B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Transactions on Circuits and Systems II Analog and Digital Signal Processing*, vol. 39, no. 11, pp. 799-814, 1992.
3. M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," *Proceedings of the 30th Conference on Design Automation*, Dallas, TX, 1993, pp. 612-616.
4. Y. P. Chen and D. F. Wong, "An algorithm for zero-skew clock tree routing with buffer insertion," *Proceedings of the European Conference on Design and Test*, Paris, France, 1996, p. 230.
5. G. E. Tellez and M. Sarrafzadeh, "Minimal buffer insertion in clock trees with skew and slew rate constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 4, pp. 333-342, 1997.
6. R. Chaturvedi and J. Hu, "Buffered clock tree for high quality IC design," *Proceedings of the 5th International Symposium on Quality Electronic Design*, San Jose, CA, 2004, pp. 381-386.
7. D. Lee and I. L. Markov, "CONTANGO: integrated optimization of SoC clock networks," *VLSI Design*, vol. 2011, article no. 407507, 2011.
8. X. W. Shih and Y. W. Chang, "Fast timing-model independent buffered clock-tree synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 9, pp. 1393-1404, 2012.
9. A. Vittal, H. Ha, F. Brewer, and M. Marek-Sadowska, "Clock skew optimization for ground bounce control," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 1996, pp. 395-399.
10. P. Vuillod, L. Benini, A. Bogliolo, and G. De Micheli, "Clock-skew optimization for peak current reduction," *Proceedings of the International Symposium on Low Power Electronics and Design*, Monterey, CA, 1996, pp. 265-270.
11. Y. T. Nieh, S. H. Huang, and S. Y. Hsu, "Minimizing peak current via opposite-phase clock tree," *Proceedings of the 42nd Design Automation Conference*, San Diego, CA, 2005, pp. 182-185.
12. R. Samanta, G. Venkataraman, and J. Hu, "Clock buffer polarity assignment for power noise reduction," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2006, pp. 558-562.
13. J. Lu and B. Taskin, "Clock buffer polarity assignment considering capacitive load," *Proceedings of the 11th International Symposium on Quality Electronic Design*, San Jose, CA, 2010, pp. 765-770.
14. P. Y. Chen, K. H. Ho, and T. T. Hwang, "Skew aware polarity assignment in clock tree," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2007, pp. 376-379.
15. H. Jang, D. Joo, and T. Kim, "Buffer sizing and polarity assignment in clock tree synthesis for power/ground noise minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 1, pp. 96-109, 2011.
16. D. Joo and T. Kim, "WaveMin: a fine-grained clock buffer polarity assignment combined with buffer sizing," *Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference*, New York, NY, 2011, pp. 522-527.
17. M. Kang and T. Kim, "Clock buffer polarity assignment considering the effect of delay variations," *Proceedings of the 11th International Symposium on Quality Electronic Design*, San Jose, CA, 2010, pp. 69-74.
18. Y. Ryu and T. Kim, "Clock buffer polarity assignment combined with clock tree generation for power/ground noise minimization," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2008, pp. 419-419.
19. J. Lu and B. Taskin, "Clock tree synthesis with XOR gates for polarity assignment," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, Lixouri, Kefalonia, 2010, pp. 17-22.
20. Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessort," *Proceedings of the Design Automation Conference*, Los Angeles, CA, 2000, pp. 168-171.
21. G. Venkataraman, Z. Feng, J. Hu, and P. Li, "Combinatorial algorithms for fast clock mesh optimization," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2006, pp. 563-567.
22. K. Jain, "A factor 2 approximation algorithm for the generalized Steiner network problem," *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, Palo Alto, CA, 1998, pp. 448-457.
23. A. Rajaram and D. Z. Pan, "MeshWorks: a comprehensive framework for optimized clock mesh network synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1945-1958, 2010.
24. Nanoscale Integration and Modeling (NIMO) Group, Arizona State University, "Predictive technology model," http://ptm.asu.edu/.
25. H. Chang and S. S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 9, pp. 1467-1482, 2005.
26. H. W. Ott, Noise Reduction Techniques in Electronic Systems, 2nd ed., New York: John Wiley & Sons, 1988.

27. D. Pandini, G. A. Repetto, and V. Sinisi, "Clock distribution techniques for low-EMI design," *Integrated Circuit and System Design: Power and Timing Modeling, Optimization and Simulation, Lecture Notes in Computer Science Volume 4644*, N. Azemard and L. Svensson, editors, Heidelberg: Springer Berlin, pp 201-210, 2007.
28. K. B. Hardin, J. T. Fessler, and D. R. Bush, "Spread spectrum clock generation for the reduction of radiated emissions," *Proceedings of the IEEE International Symposium on Electromagnetic Compatibility,* Chicago, IL, 1994, pp. 227-231.
29. X. Hu and M. R. Guthaus, "Clock tree optimization for electromagnetic compatibility (EMC)," *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, Yokohama, Japan, 2011, pp. 184-189.
30. Y. S. Su, W. K. Hon, C. C. Yang, S. C. Chang, and Y. J. Chang, "Value assignment of adjustable delay buffers for clock skew minimization in multi-voltage mode designs," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2009, pp. 535-538.
31. A. Kapoor, N. Jayakumar, and S. P. Khatri, "A novel clock distribution and dynamic de-skewing methodology," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2004, pp. 626-631.
32. K. H. Lim and T. Kim, "An optimal algorithm for allocation, placement, and delay assignment of adjustable delay buffers for clock skew minimization in multi-voltage mode designs," *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, Yokohama, Japan, 2011, pp. 503-508.
33. S. H. Huang, C. M. Chang, and Y. T. Nieh, "State re-encoding for peak current minimization," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2006, pp. 33-38.
34. L. Benini and G. De Micheli, "State assignment for low power dissipation," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 258-268, 1995.
35. Y. Lee and T. Kim, "State encoding algorithm for peak current minimization," *IET Computer & Design Techniques*, vol. 5, no. 2, pp. 113-122, 2011.
36. F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah, "PBS: a backtrack-search pseudo-boolean solver and optimizer," *Proceedings of the 5th International Symposium on Theory and Applications of Satisfiability Testing*, Cincinnati, OH, 2000, pp. 346-353.
37. J. Gu, G. Qu, L. Yuan, and Q. Zhou, "Peak current reduction by simultaneous state replication and re-encoding," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design,* San Jose, CA, 2010, pp. 592-595.
38. E. H. Nam, K. S. Choi, J. Y. Choi, H. J. Min, S. L. Min, "Hardware platforms for flash memory/NVRAM software development," *Journal of Computer Science and Engineering*, vol. 3, no. 3, pp. 181-194, 2009.
39. T. Kim, "Task-level dynamic voltage scaling for embedded system design," *Journal of Computer Science and Engineering*, vol. 4, no. 3, pp. 189-206, 2010.

Deokjin Joo

Deokjin Joo received the B.S. degree and the M.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2009 and 2011, respectively. He is currently working toward the Ph.D. degree in the School of Electrical Engineering and Computer Science, Seoul National University. His current research interests include clock tree synthesis for low power and thermal resilient design.



Minseok Kang

Minseok Kang received the B.S. and M.S. degrees in electrical engineering at Seoul National University, Korea in 2008 and 2010, respectively. Currently he is pursuing Ph.D. degree in electrical engineering at Seoul National University, Korea. His research interests are in clock mesh synthesis and variation-aware design.

Taewhan Kim

Taewhan Kim received the B.S. degree in computer science and statistics and the M.S. degree in computer science from Seoul National University, Seoul, Korea, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, in 1993. He is currently a Professor with the School of Electrical Engineering and Computer Science, Seoul National University. After graduation, he was with Lattice Semiconductor Corporation and Synopsys, Inc., San Jose, CA, for six years, specializing in design automation tool development. He has published over 160 technical papers in international journals and conferences, including the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, the IEEE Transactions on Very Large Scale Integration, ACM TODAES, DAC, ICCAD, and ASPDAC. His current research interests include computer-aided design of integrated circuits ranging from the architectural synthesis through physical designs, specifically focusing on power, thermal, noise, reliability, and 3-D IC design issues.