

Clock Network Synthesis with Concurrent Gate Insertion

Jingwei Lu, Wing-Kai Chow, and Chiu-Wing Sham

The Hong Kong Polytechnic University
francesco.ljw@gmail.com,
williamchowhk@gmail.com,
encwsham@eie.polyu.edu.hk

Abstract. In VLSI digital circuits, clock network plays an important role on the total performance of the chip. Clock skew and power dissipation are two major focuses of concerns in the clock network synthesis. During topology generation, the locations of buffer and gate insertion are usually not available. Despite local optimization, the global performance is limited. In this paper, a novel approach of topology generation with concurrent gate insertion is proposed. Meanwhile, a strict clock slew constraint is applied with comprehensive buffer insertion techniques. By clock gating, the switched capacitance of the clock tree is reduced, with acceptable extra cost caused in controller tree. In experimental results it is shown that our approach has good performance on the reduction of both clock skew and power dissipation.

1 Introduction

Clock signals are employed in VLSI digital systems to synchronize the active components of a design. Clock skew minimization is a popular research topic during the past decades. Some early works [1,2] mainly concentrated on the average distribution of wirelength between source and each terminal to achieve actual delay equalization. Afterwards, delay balancing [3] using Elmore delay model [4] became prevalent to acquire more accurate information of timing delay. The deferred-merging and embedding (DME) technique was proposed in [5], it can achieve the zero clock skew with minimal wirelength. In topology generation, some algorithms were proposed for unbuffered and ungated clock tree in [6], and buffered but ungated clock tree in [7]. In ISPD 2009 clock network synthesis contest [8], a voltage variation related objective named Clock Latency Range (CLR) was formulated. Subsequent research work was also proposed [9] accordingly.

Twenty percent to fifty percent of the power usage is contributed by the clock network [10]. On behalf of power reduction, the application of clock gating is an effective approach in the sequential circuits. The principal idea is to turn off the idle modules and tree sections in order to cut down the unnecessary switching power. Clock gating can be applied on logic level [11], register-transfer-level level [12] and architecture level [13]. Nevertheless, besides logical information, physical location of the modules should also be taken into account in case wirelength overhead thus power usage waste. Some achievements were proposed with both logical and physical concerns. The algorithm in [14] showed a clock tree topology construction, taking advantage of the activity patterns of modules. Moreover, activity similarity was considered in [15]. Besides,

a gating method regarding microprocessor design was proposed in [16]. The algorithm constructed the topology in a bottom-up procedure, with the objective of switched capacitance minimization. Further on, in [17] a comprehensive technique with a recursive computation on effective switched capacitance and a solution sampling on merging segment set was discussed.

In this paper, we propose a novel synthesizer to construct a binary clock tree in a bottom-up course. Simultaneous optimization on the clock skew and the power dissipation is applied. The topology generator is responsible for a buffered and gated clock tree, and the clock gates are inserted concurrently. The major advantage of our work is to take the downstream masking information of subtrees into account during each merging step. An algorithm named dual-MST [9] for topology generation is involved in our work, and the cost function is improved for power awareness. Besides, we perform a more strict slew constraint along the whole clock network. Thus the constraint on buffer and gate location is emphasized. The experimental results show that our method can greatly reduce the power consumption of the clock network with proper gate insertion. Meanwhile, the clock skew and PVT variation can still be maintained within an acceptable range.

The rest of the paper is organized as follows. Some preliminary knowledges of tree construction and capacitance are discussed in section 2. The details of our approach are discussed in section 3. The technique of power aware topology generation with concurrent buffer and gate insertion is proposed detailedly. Experimental results are shown in section 4. Finally we reach our conclusion in section 5.

2 Preliminaries

2.1 Clock Tree and Controller Tree

Let $T = \{V, E\}$ denote the clock tree. $V = \{v_i | i = 1, 2, \dots, m_v\}$ is the set of nodes, and $E = \{e_j | j = 1, 2, \dots, m_v - 1\}$ is the set of clock edges between the node v_j and its corresponding parent. Let $|e_j|$ denote the length of the edge e_j . Apparently, for the root node there will be no edge assigned. Let $G = \{g_i | i = 1, 2, \dots, m_v - 1\}$ denote the set of gates. The gate g_j is assigned to be on the edge e_j masking the node v_j directly. We use $S = \{v_k | k = 1, 2, \dots, m_s\}$ (where $m_s < m_v$) to denote the set of modules (the sinks, or leaf nodes). The rest ($m_v - m_s$) nodes are named internal nodes. The root is said to be at level 0. Node v_i is said to be at level n_i if there are n_i edges on the path from v_i to the root of the tree. Moreover, we assume that the topology of the clock tree is full binary. Every internal node has exactly two children. The skew of T is the difference between the longest signal delay and the shortest signal delay from the source to any sinks. As proposed in [16], we assume that the control logic is located at the center of the chip. Star routing is also applied in the controller tree, denoted as T^{ctr} . A control edge EN_i in T^{ctr} will transmit the enable signal to the respective gate g_i on the edge e_i in the clock tree T . An example of a clock tree T as well as its controller tree T^{ctr} is shown in figure 1.

During the operating time of a circuit, each module will have its active and idle times. It is usually specified as different activity patterns. The activity patterns can be obtained by the simulation of the design at the behavioral level [14]. Let A_i denote the activity

pattern of the node v_i . It is a binary string with 1s indicating the active periods and 0s indicating the idle periods of a sink or an internal node. If v_i is a sink node, we can directly obtain A_i from the benchmark file. Otherwise, suppose v_i to be an internal node with two children nodes v_L and v_R accordingly. The clock signal at v_i must be enabled whenever its left or right child is active. Therefore, A_i is calculated by performing the bitwise OR operation on the activity patterns of v_L and v_R . Hence $A_i = A_L \cup A_R$. An example of a bottom-up activity pattern transmission is shown in figure 2.

Let $P(A_i)$ denote the activity of the node v_i , and $P_{tr}(A_i)$ denote its transition probability. These two factors are calculated based on the corresponding pattern of v_i . The specific equations are shown as below

$$P(A_i) = \frac{AT_{no}(A_i)}{Len(A_i)}, P_{tr}(A_i) = \frac{TR_{no}(A_i)}{2 \times (Len(A_i) - 1)} \tag{1}$$

where $AT_{no}(A_i)$ is the number of active times (1s) in A_i , and $TR_{no}(A_i)$ is the number of transitions (10 or 01) in A_i . $Len(A_i)$ denotes the stream length of A_i .

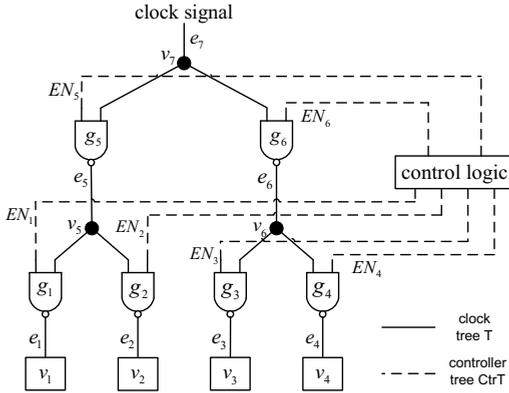


Fig. 1. A gated clock binary tree

2.2 Switched Capacitance

The power consumed by CMOS circuits consists of two components: static and dynamic power. The static power is mostly determined by the feature size and other technology. Therefore, in this paper we only consider dynamic power minimization. The definition of the dynamic power is $P = \frac{1}{2} \alpha C f V_{dd}^2$. C means the total load capacitance on the circuit, f is the frequency of the clock signal and V_{dd} is the power supply. α means the amount of switch times in each clock cycle. For clock tree $\alpha = 2$, because there is one rising and one falling edge in each clock period. $\alpha = 1$ in the controller tree, respectively. Since f and V_{dd} are constant parameters in the digital circuits, we can use the switched capacitance as a measure of the power usage. Assume that a subtree T_i rooted at v_i with a gate insertion g_i , and the controller tree is denoted as T_i^{ctr} . The unmasked load capacitance for T_i and T_i^{ctr} are $C_{v_i}^u$ and $C_{T_i^{ctr}}^u = C_{EN_i} + C_g$ accordingly, C_g denotes the input capacitance of a gate. We can get the equation for the

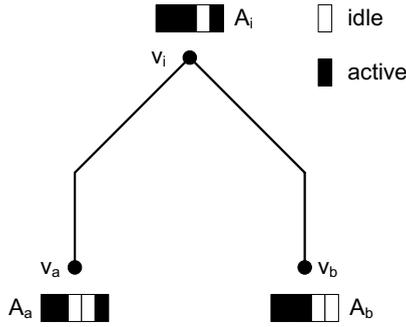


Fig. 2. An example of activity pattern transmission

downstream switched capacitance of v_i as $SC_{v_i} = C_{v_i}^u P(A_i)$. Similarly, the corresponding switched capacitance for the controller tree T_i^{ctr} is measured as $SC_{T_i^{ctr}} = (C_{EN_i} + C_g) P_{tr}(A_i)$.

The power consumption of a clock network is directly proportional to the average switched capacitance for each clock cycle. The total switched capacitance is contributed by a gated and buffered clock tree T and a controller tree T^{ctr} . In order to reduce the switching activity, modules and clock tree sections can be disabled by clock gates during their inactive clock periods. From the above example, we can see that the original capacitance of node v_i is $C_{v_i}^u$. With gate g_i inserted at v_i , the resultant switched capacitance is $SC_{v_i} + SC_{T_i^{ctr}}$. If $C_{v_i}^u < SC_{v_i} + SC_{T_i^{ctr}}$, the capacitance will be reduced with the insertion of g_i . A power aware clock tree topology with proper buffer and gate insertion will efficiently reduce the switched capacitance, hence cut down the power usage of the circuits. Given the physical location of the modules together with the models of wire, buffer and gate, the objective of our work is to construct a buffered and gated clock network and a controller network. Subject to the two constraints of nominal zero skew and maximal slew rate, the average switched capacitance should be minimized.

3 Methodology

We build our clock tree based on the dual-MST construction method [9], and the resulting clock tree is close to a full symmetry. In our paper, it is improved with a new cost function to take both distance and power saving into account. As a result, the according topology can result in both low power usage and small clock skew. A recursive buffer/clock gate insertion method is developed for bottom-up merging. Blockage handling technique is also involved, because the buffers and gates cannot be placed inside blockage regions. Elmore model [4] is applied for clock delay computation. DME technique [18] is applied for wirelength minimization. Thus, segment is used instead of point to represent the set of merging location, and deferred embedding is applied to reduce total wirelength.

3.1 Power Aware Topology Generation

In order to save the power, the nodes with a bigger similarity of activity patterns should have a higher priority to be matched. Assume v_a and v_b to be a pair of two nodes,

as shown in figure 2. If the corresponding activity patterns A_a and A_b are similar, the resulted activity A_i will have a shorter active period, and smaller power cost will be caused. Besides the concerns on activity patterns, an estimation of the merging cost $Pwr(v_a, v_b)$ is also required. This can be determined in multiple ways. For instance, we can actually merge the two nodes together to obtain the exact connection information. However, exact buffer insertion and wire balancing are performed, which will cost longer time. Instead, we develop a new method for potential switched capacitance estimation. The Manhattan distance between the nodes v_a and v_b is denoted by $D(v_a, v_b)$. The Elmore delay difference of these two nodes is denoted by $DLY(v_a, v_b)$. The delay and power consumption for unit wirelength are denoted by ρ_D and ρ_P respectively, which are computed in advance for simulation reference. If $\frac{DLY(v_a, v_b)}{\rho_D}$ is smaller than $D(v_a, v_b)$, then the two nodes can be merged without snaking wire involved, and the corresponding equation for power cost computation is shown as below

$$Pwr(v_a, v_b) = \rho_P \times D(v_a, v_b) \times P(A_i) \quad (2)$$

Otherwise, snaking will be included, as shown in the following equation

$$Pwr(v_a, v_b) = \rho_P \times \frac{DLY(v_a, v_b)}{\rho_D} \times P(A_i) \quad (3)$$

An improved power aware dual-MST geometric matching technique is developed for topology construction, a specific definition of a geometric matching of one iteration can be found in [2]. The detailed description is shown in procedure 1. It is a weighted perfect matching approach. Given a set of nodes $V = \{v_1, v_2 \dots v_m\}$, we first construct a complete graph $G = \{V, E\}$. Let $|V|$ and $|E|$ denote the number of nodes and edges in the graph G , so $|V| = m$. Since G is a complete graph, every pair of two nodes v_i, v_j is connected by an edge $e_{i,j}$, $E = \{e_{1,2}, e_{1,3} \dots e_{m-1,m}\}$ and $|E| = \frac{m(m-1)}{2}$. The cost of matching two nodes v_i and v_j is denoted as $f_c(e_{i,j})$. Let M denote the matching result of G . M is composed of a group of edges and it is a subset of E . The maximal pairing cost of M is denoted as C_{max} and defined as below. We will get close to a symmetric clock tree by reducing C_{max} in each level. The merging cost $f_c(v_a, v_b)$ is shown as below. α and β are the weight of the Manhattan distance and the estimated power cost, respectively.

$$f_c(v_a, v_b) = \alpha \times D(v_a, v_b) + \beta \times Pwr(v_a, v_b) \quad (4)$$

By means of this weighted cost function, the node pairs with a bigger similarity of switching activity and a shorter distance will have a higher priority to be matched. Our approach of topology generation is based on concurrent gate insertion, therefore the downstream information of the two merging nodes are accurate.

3.2 Concurrent Gate and Buffer Insertion

A recursive buffer and gate insertion technique is developed on behalf of three objectives: (1) slew rate constraint (2) clock skew minimization (3) power usage reduction. Buffers are utilized for power supply to restrict the signal transition time, and clock

Procedure 1. Partition(G)

Require: $G = \{V, E\}$ is a complete graph, E is sorted in ascending order of $f_c(e_{i,j})$.

```

if  $|V| \leq 1$  then
  return;
else if  $|V| = 2$  then
  merge( $v_1, v_2$ );
  return;
else
  Building dual-MST with  $|V| - 2$  edges inserted.
  Two subgraphs  $G' = \{V', E'\}$  and  $G'' = \{V'', E''\}$  are generated
  Two minimum spanning trees  $st'$  and  $st''$  for  $V'$  and  $V''$  are generated
  if  $|V'|$  is odd and  $|V''|$  is odd then
     $e_{m,n} = \arg_{e_{i,j}} \min\{f_c(e_{i,j}) \mid \forall n_i \in V', \forall n_j \in V''\}$ ;
    merge( $v_m, v_n$ );
    remove  $v_m$  from  $V'$ ;
    remove  $v_n$  from  $V''$ ;
    remove  $e_{m,x}$  from  $E', \forall x \in V'$ ;
    remove  $e_{n,y}$  from  $E'', \forall y \in V''$ ;
  end if
  partition( $G'$ );
  partition( $G''$ );
  return;
end if

```

gate insertion can reduce the switched capacitance by disabling idle sections. Real-time simulation of signal slew rate costs much more time and is impractical. Hence we build look-up tables in advance for slew reference. It can estimate the driving ability among diverse circumstances. We model the buffer and gate with according attributes for Elmore delay computation. Some previous works [19] already proposed to construct a buffered clock tree with zero clock skew. In our work, we apply similar approach for both buffer and clock gate insertion. The input/output capacitance and resistance of the buffers and clock gates should be obtained first. Hence, the delay of wire, buffers and clock gates can be computed based on Elmore RC model.

In our work, we try to maintain the level of buffers and gates of every source-to-sink clock path exactly the same. During the procedure of the bottom-up binary merging, we first examine the two downstream levels of gates. If they differ by two or more, a penalty cost will be engaged. Such matching result will probably be discarded due to the huge cost. Buffer levels will be balanced accordingly. By means of this level balancing, the clock skew will be reduced significantly, and the negative effect caused by signal variation will be reduced.

Here we will describe our technique of gate insertion based on a determined matching result. We first define three different kinds of gate insertion. They are virtual gate insertion at the upstream level, temporal gate insertion at the current level and none gate insertion. Temporal insertion is controlled by the balancing of gate levels, which will be further divided into two kinds of single gate insertion and one kind of back-to-back double gates insertion. The insertion of a gate is assumed to be closest to the internal

merging node on behalf of switched capacitance minimization. Since DME technique is applied in our work, we assume the middle point of the merging segment to be the gate location. The comparison among the three assumption of gate insertion are based on the resulting switched capacitance, which are SC_{vir} , SC_{tmp} and SC_{non} , respectively. If the power consumption of the virtual insertion or the none insertion is the smallest, no insertion of any gate will definitely result in less switched capacitance compared to the choice of temporal gate insertion. Therefore, we discard any insertion of gates at the current level. Otherwise, temporal gate insertion will probably reduce the switched capacitance rather than the others, and here we will accept the insertion of gates.

An example is shown in figure 2. The activity A_i equals to $A_a \cup A_b$. The edge connection between each of the two nodes to the merging node are denoted as e_a and e_b . C_{e_a} and C_{e_b} are their corresponding capacitance cost. The equations to compute the three resulting switched capacitance are shown as below

$$SC_{vir}(v_a, v_b) = (C_a^u + C_{e_a} + C_b^u + C_{e_b}) \times P(A_i) + C_{T_{ctr}^u} \times P_{tr}(A_i) \quad (5)$$

$$SC_{tmp}(v_a, v_b) = (C_a^u + C_{e_a}) \times P(A_a) + C_{T_a^{ctr}^u} \times P_{tr}(A_a) + C_b^u + C_{e_b} \quad (6)$$

$$SC_{non}(v_a, v_b) = C_a^u + C_{e_a} + C_b^u + C_{e_b} \quad (7)$$

Notice that here we only describe the equation of SC_{tmp} for a single gate insertion at node v_a . The other two equations can be derived in a similar way.

4 Experimental Results

In this section, our experimental results are presented. We implement our clock network synthesizer in C programming language. The binary is executed on a Linux machine with an Intel Core2 Quad 2.4G Hz CPU and 4GB memory. The benchmark circuits used in the experiments are released from the ISPD 2009 CNS contest [8]. The detailed information of the benchmark circuits is shown in table 2. In our experiment, one type of wire and one type of buffer is used in our clock tree synthesizer. The unit resistance of the wire is $0.0003\Omega/nm$, and the unit capacitance of the wire is $0.00016fF/nm$. The specific configuration of the buffer in different sizes is shown in table 1. In our synthesizer, the maximum buffer size is set to be 6. Hence we list the attributes of different buffer sizes up to 6. Notice that the corresponding attributes of a gate is listed in the last row of table 1. This table is generated from our SPICE simulation statistics. C_b means the input capacitance, R_b means the driver resistance and d_b means the internal delay of a buffer, respectively. During the evaluation, the power supply is set to be $V_{dd} = 1.0V$. The PTM model applied in our simulation are of 45 nanometer scale.

A summary of the performance of our clock tree after insertion of clock gates is shown in table 3. We run our program with different values of α and β for topology tuning. The clock skew (SKEW), total capacitance (TC), optimal capacitance (OSC), switched capacitance (SC) and CPU time are listed. The respective units are picoseconds (ps) for SKEW, seconds for CPU and femto-farad (fF) for capacitance. TC denotes the

Table 1. Buffer configuration

buffer sizes	C_b (fF)	R_b (Ω)	d_b (ps)
1	35	66.9	4.92
2	70	40.5	5.63
3	105	31.3	6.13
4	140	26.4	6.52
5	175	25.0	6.95
6	210	20.7	7.20
gate	35	52.45	17.03

Table 2. Circuit information of the benchmarks from ISPD 2009

Circuits	Chip Size (mm x mm)	No. of sinks	No. of block (Area %)	limit CAP (fF)
ispd09f11	11.0 x 11.0	121	0 (0%)	118000
ispd09f12	8.1 x 12.6	117	0 (0%)	110000
ispd09f21	12.6 x 11.7	117	0 (0%)	125000
ispd09f22	11.7 x 4.9	91	0 (0%)	80000
ispd09f31	17.1 x 17.1	273	88 (24.38%)	250000
ispd09f32	17.0 x 17.0	190	99 (34.26%)	190000
ispd09fnb1	2.6 x 2.1	330	53 (37.69%)	42000
ispd09f33	15.3 x 15.3	209	80 (27.68%)	195000
ispd09f34	16.0 x 16.0	157	99 (38.67%)	160000
ispd09f35	15.3 x 15.3	193	96 (33.22%)	185000
ispd09fnb2	6.4 x 4.4	440	1346 (63.88%)	88000
avg.	12.1 x 11.6	203	169 (23.62%)	140273

original total capacitance cost of the clock tree without gate insertion. OSC denotes the resulted capacitance after disabling of all the idle periods at each node. SC denotes the resulted switched capacitance of our gated clock tree. It can be seen that SC is mostly smaller than TC , which means a effective power reduction in our gated clock tree construction. The nominal skew of each clock tree is zero. Additionally, we use NGSPICE for further evaluation and get the accurate skew estimation, as listed in the table. The activity pattern of all the sinks are generated according to the instruction and RTL description used in [16]. The length of the activity pattern is 10000 for every benchmark. Previous works were proposed with loose constraint on slew or driving power supply, for instance, $\leq 20 \times C_g$ for a buffer or gate insertion in [17,16]. The work in [14] did not involve clock routing and synthesis. However, in our program the transition time (slew rate) is maintained to be under 100 ps throughout the whole clock network, thus more buffers are inserted to follow this rule. As a matter of fact, in this paper it is very difficult for us to include direct comparison with previous works. It can be speculated that the power cost of our work should be larger than the previous ones, but the signal transition time is more consistent hence the work is more practical in use. Generally, in our work the switched capacitance can be reduced by around 10% with the insertion of clock gates. Meanwhile, the clock skew is only about 20 ps in average. The runtime of our program is less than 3 seconds, which represents good efficiency.

Table 3. Clock skew and switched capacitance with gate insertion

Circuits	Our approach ($\alpha = 1, \beta = 0$)					Our approach ($\alpha = 2, \beta = 1$)				
	SKEW	TC	OSC	SC	CPU	SKEW	TC	OSC	SC	CPU
ispd09f11	20.0	103973	61868	78939	0.37	16.7	103851	61422	78261	0.37
ispd09f12	17.2	104874	65539	78970	0.34	16.6	103998	65090	79603	0.35
ispd09f21	20.0	118028	68813	89140	0.35	25.7	108116	67586	81043	0.35
ispd09f22	15.6	69810	43786	53173	0.32	8.5	69552	43938	53597	0.32
ispd09f31	33.7	221639	136596	179336	3.83	19.3	220522	128744	174024	5.60
ispd09f32	33.4	175122	101850	138156	0.51	21.7	162525	103658	123151	0.50
ispd09f33	20.6	171747	107773	139467	5.44	18.8	155995	100329	128386	6.30
ispd09f34	22.2	144688	92341	118570	0.49	20.3	139518	88924	109183	0.46
ispd09f35	16.9	165546	104232	134708	8.11	21.6	163376	102231	128963	8.13
ispd09fnb1	18.6	32635	23452	32635	0.70	29.6	34370	24869	34370	0.63
ispd09fnb2	19.7	67041	46550	66280	2.40	27.5	70478	50113	69788	1.90
avg.	21.6	125009	77527	100852	2.08	20.6	121118	76082	96397	2.26

5 Conclusion

In conclusion, power saving and clock skew are two major concerns in clock network synthesis. A power aware topology generation with concurrent buffer/gate insertion is proposed in this paper. This is developed in order to optimize the clock skew and the power dissipation of a clock distribution network simultaneously. Experimental results show that our method can greatly reduce the switched capacitance hence power consumption of the clock network with proper clock gate insertion. Meanwhile, the clock skew can still be maintained within an acceptable range.

Acknowledgement

The work described in this article was partially supported by the RGC Direct Allocation Fund from The Hong Kong Polytechnic University (Project No. A-PC0W).

References

1. Jackson, M.A.B., Srinivasan, A., Kuh, E.S.: Clock Routing for High-Performance ICs. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 573–579 (June 1990)
2. Kahng, A., Cong, J., Robinsh, G.: High-Performance Clock Routing Based on Recursive Geometric Matching. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 322–327 (June 1991)
3. Tsayz, R.S.: Exact Zero Skew. In: Proceedings of IEEE/ACM International Conference on Computer Aided Design, pp. 336–339 (November 1991)
4. Elmore, W.C.: The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics* 19(1), 55–63 (1948)
5. Boese, K.D., Kahng, A.B.: Zero-Skew Clock Routing Trees With Minimum Wirelength. In: Proceedings of 5th the Annual IEEE International ASIC Conference and Exhibit, pp. 17–21 (1992)

6. Edahiro, M.: A Clustering-Based Optimization Algorithm in Zero-Skew Routings. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 612–616 (June 1993)
7. Chaturvedi, R., Hu, J.: Buffered Clock Tree for High Quality IC Design. In: Proceedings of the International Symposium on Quality Electronic Design, pp. 381–386 (2004)
8. Sze, C.N., Restle, P., Nam, G.-J., Alpert, C.: ISPD 2009 Clock Network Synthesis Contest. In: Proceedings of ACM International Symposium on Physical Design, pp. 149–150 (March 2009)
9. Lu, J., Chow, W.K., Sham, C.W., Young, E.F.Y.: A Dual-MST Approach for Clock Network Synthesis. In: Proceedings of Asia and South Pacific Design Automation Conference, pp. 467–473 (January 2010)
10. Kitahara, T., Minami, F., Ueda, T., Usami, K., Nishio, S., Mruakata, M., Mitsuhashi, T.: A Clock-Gating Method for Low-Power LSI Design. In: Proceedings of Asia and South Pacific Design Automation Conference, pp. 307–312 (January 1998)
11. Chang, C.M., Huang, S.H., Ho, Y.K., Lin, J.Z., Wang, H.P., Lu, Y.S.: Type-Matching Clock Tree for Zero Skew Clock Gating. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 714–719 (June 2008)
12. Donno, M., Ivaldi, A., Benini, L., Macii, E.: Clock-Tree Power Optimization based on RTL Clock-Gating. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 622–627 (June 2003)
13. Luo, Y., Yu, J., Yang, J., Bhuyan, L.: Low Power Network Processor Design Using Clock Gating. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 712–715 (June 2005)
14. Farrahi, A.H., Chen, C., Srivastava, A., Tellez, G., Sarrafzadeh, M.: Activity-Driven Clock Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20(6), 705–714 (2001)
15. Chen, C., Kang, C., Sarrafzadeh, M.: Activity-Sensitive Clock Tree Construction for Low Power. In: International Symposium on Low Power Electronics and Design, pp. 279–282 (2002)
16. Oh, J., Pedram, M.: Gated Clock Routing for Low-Power Microprocessor Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20(6), 715–722 (2001)
17. Chao, W.C., Mak, W.K.: Low-Power Gated and Buffered Clock Network Construction. *ACM Transactions on Design Automation of Electronic Systems* 13(1) (January 2008)
18. Chao, T.H., Hsu, Y.C., Ho, J.M.: Zero Skew Clock Net Routing. In: Proceedings of IEEE/ACM Design Automation Conference, pp. 518–523 (July 1992)
19. Chen, Y.P., Wong, D.F.: An Algorithm for Zero-Skew Clock Tree Routing with Buffer Insertion. In: European Design and Test Conference, pp. 230–236 (March 1996)