Clock Buffer Polarity Assignment for Power Noise Reduction*

Rupak Samanta, Ganesh Venkataraman, Jiang Hu
Department of ECE, Texas A&M University, College Station, Texas 77843
rupak9@tamu.edu, ganesh@ece.tamu.edu, jianghu@ece.tamu.edu

ABSTRACT

Power/ground noise is a major source of VLSI circuit timing variations. This work aims to reduce clock network induced power noise by assigning different signal polarities (opposite switchings) to clock buffers in an existing buffered clock tree. Three assignment algorithms are proposed: (1) partitioning, (2) 2-coloring on minimum spanning tree and (3) recursive min-matching. A post-processing of clock buffer sizing is performed to achieve desired clock skew. SPICE based experimental results indicate that our techniques could reduce the average peak current and average delay variations by **44%** and **54%** respectively.

1. INTRODUCTION

When the supply voltage decreases with VLSI technology scaling, circuit performance becomes increasingly vulnerable to power/ground noise [1, 2]. Based on an estimation in [3], a $0.1V$ power noise may cause 80% inverter delay variation at $45nm$ technology. A main culprit of power noise is clock network which keeps drawing huge current frequently from the power supply network [4, 5]. In order to reduce the clock-induced power noise, some works [4] attempt to avoid simultaneous flip-flop switchings through clock skew scheduling.

Besides the flip-flops, the switchings of clock buffers also contribute greatly to the clock-induced noise. In a clock network of an industrial ASIC design, there could be several thousands of clock buffers [6]. A recent work [5] proposes to use different signal polarities on clock buffers so that the roughly simultaneous same-direction switchings are replaced by a mixture of opposite-direction switchings. Signal polarity refers to whether or not a signal switches in the same direction as the clock source. The main idea of [5] is illustrated in Figure 1. In Figure 1(a), all buffers have the same signal polarity and therefore they have either simultaneous rising switches, which draw large current from power (V_{dd}) network, or simultaneous falling switches which draw large current from ground (V_{ss}) network. In contrast, the application of opposite polarities as in Figure 1(b) decreases current withdraw since only a half of the buffers draw current from V_{dd} while the others draw from V_{ss} at the same time. Please note that polarity assignment to a buffer is different from selecting between inverting or non-inverting type for the buffer, although these two are related. By using different types of flip-flops, positive-edge or negative-edge triggered, both signal polarities can be accommodated at flip-flops with hardly any impact to the original circuit design.

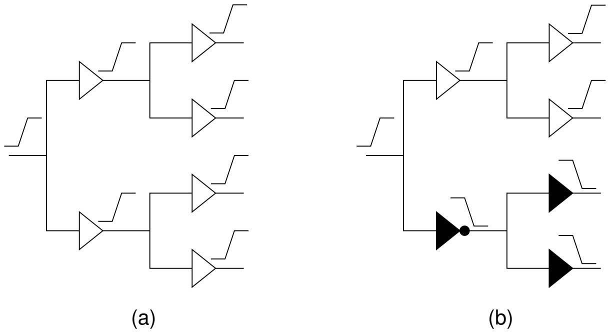

Figure 1: All buffers in (a) have positive signal polarity and switch in the same direction. The dark buffers in (b) are assigned with negative polarity and switch in the direction opposite to the buffers with positive polarity.

When assigning polarities, the work of [5] partitions the clock sinks (flip-flops) into two subsets, one for positive polarity and the other for negative polarity. Then, two subtrees are constructed separately for the two subsets, i.e., one subtree has only positive polarity and the other subtree has only negative polarity. However, this approach faces a dilemma considering the following two typical scenarios:

- If the two subsets are spatially separated from each other like in Figure 2(a), the two subtrees (one in solid lines and the other in dashed lines in Figure 2) are in two separated regions. Except the boundary region between the two subtrees, the power noise in a local area such as the shaded regions in Figure 2(a), is not reduced by the application of opposite polarities. This is because that power noise is mostly a local effect.

*This work is partially supported by Semiconductor Research Corporation under contract 2004-TJ-1205.

ICCAD '06, November 5-9, 2006, San Jose, CA
Copyright 2006 ACM 1-59593-389-1/06/0011 ...$5.00.

- If the sink locations of the two subsets are intermingled, the approach of [5] results in two intermingled subtrees like Figure 2(b). In this scenario, the power noise in each local region can be reduced, but the wirelength of the clock network is increased greatly.

Therefore, constructing two subtrees independently [5] either is ineffective for reducing local power noise or suffers from huge wirelength overhead. Moreover, the work of [5] evaluates only the peak current while neither power supply voltage noise nor the impact on delay variation is discussed.

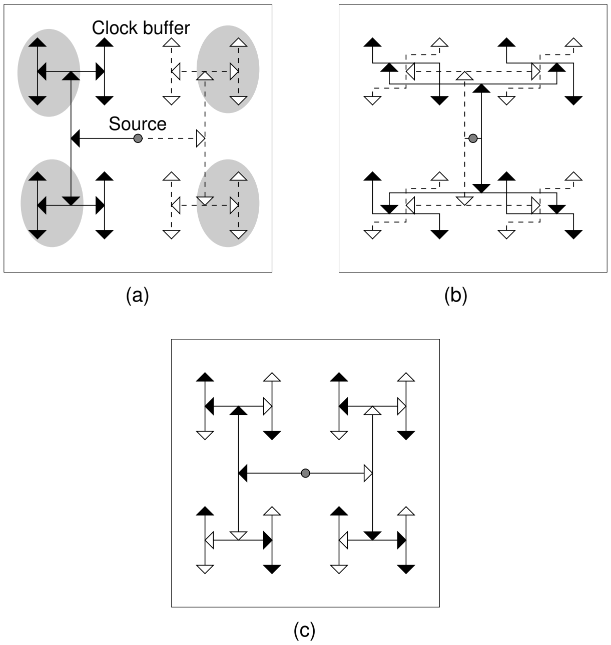

Figure 2: Constructing two subtrees separately for opposite polarities either cannot reduce local power noise if the two subtrees are spatially apart like in (a), or results in huge wirelength overhead as in (b). We propose to perform fine-grained polarity assignment on an existing clock tree as in (c).

In this paper, we propose to perform fine-grained clock buffer polarity assignment on an existing clock tree. We carry out a buffer type matching on the resulting clock tree to minimize the path unbalance that may arise due to buffer polarity assignment. Then, a clock buffer tuning is carried out to restore the clock skew altered by the polarity assignment. Three assignment algorithms are developed: (1) partitioning, (2) 2-coloring on minimum spanning tree and (3) recursive min-matching. The fine granularity of the assignment implies that even a very small region usually contains opposite polarities as long as there are more than one clock buffers. By doing so, the clock-induced power noise can be reduced almost everywhere.

2. IMPACT TO DELAY VARIATION

Power/ground noise directly affects gate/buffer delay variation [2]. We present a first order analysis on the impact of clock buffer polarity assignment to gate/buffer delay variations. Without polarity assignment, i.e., with identical polarity for all clock buffers, all clock buffers have either simultaneous rising switchings, which cause decreased V_{dd} and almost no disturbance to V_{ss} (Figure 3(a)), or simultaneous falling switchings, which raise V_{ss} but have negligible influence on V_{dd} (Figure 3(b)). With polarity assignment, both V_{dd} and V_{ss} degrade but with less degree (Figure 3(c)).

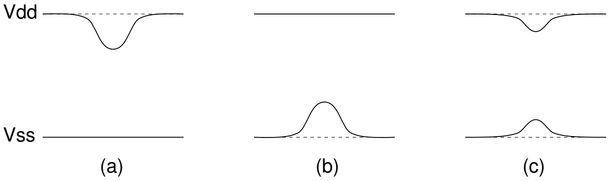

Figure 3: Power noise in a local region when (a) all buffers have rising switches, (b) all buffers have falling switchings, and (c) half of the buffers rising while the others falling.

We define power noise ΔV_{dd} and ground noise ΔV_{ss} as

$$\Delta V_{dd} = \tilde{V}_{dd} - V_{dd} \qquad \Delta V_{ss} = \tilde{V}_{ss} - V_{ss}$$

where V_{dd} and V_{ss} are ideal voltage values, and $\tilde{V}_{dd}$ and $\tilde{V}_{ss}$ are the actual voltages considering noise. As in [2], the power/ground noise can be equivalently evaluated by *differential mode noise*

$$\Delta V_{dif} = \Delta V_{dd} - \Delta V_{ss}$$

and *common mode noise*

$$\Delta V_{com} = \Delta V_{dd} + \Delta V_{ss}$$

Table 1: An example of power/ground noise for the three cases in Figure 3.

Case in Figure 3	ΔV_{dd}	ΔV_{ss}	ΔV_{dif}	ΔV_{com}
All rising (a)	-0.2	0	-0.2	-0.2
All falling (b)	0	0.2	-0.2	0.2
Half rising, half falling (c)	-0.1	0.1	-0.2	0

In Table 1, we list a rough numerical example of power/ground noise for the three cases in Figure 3. One can see that the polarity assignment does not change the differential mode noise but can reduce the common mode noise to nearly zero. According to [2], the variation of rising delay and falling delay can be expressed as

$$\Delta t_{rise} = -A \cdot \Delta V_{com} - B \cdot \Delta V_{dif}$$

and

$$\Delta t_{fall} = C \cdot \Delta V_{com} - D \cdot \Delta V_{dif}$$

respectively, where A, B, C and D are all positive constants. The three cases in Figure 3 result in approximately the same negative value of ΔV_{dif} which contributes to roughly the same amount of delay increase. The case of Figure 3(a) has more rising delay increase and less falling delay increase due to its negative common mode noise. Symmetrically, the case of Figure 3(b) has less rising delay increase and more falling delay increase. In contrast, the common mode noise from the case of Figure 3(c) is almost zero and therefore does not contribute to the delay variation. Hence, clock buffer polarity assignment, which corresponds to Figure 3(c), can reduce the worst case delay variation compared to using identical polarity.

3. PROBLEM FORMULATION

Given a buffered clock tree with n buffers, assign either positive or negative signal polarity to every buffer such that the difference between the number of positive buffers and the number of negative buffers is no greater than 1 in any region of arbitrary size.

For a region including all of the clock buffers, this objective requires that roughly a half of the buffers have positive polarity and the others have negative polarity. For a small region containing only two clock buffers, this formulation requests one of them is positive and the other is negative.

4. POLARITY ASSIGNMENT ALGORITHMS

We propose three heuristic algorithms to solve the problem formulated in the previous section.

4.1 Partitioning

First, a graph $G = (V, E)$ is constructed with each node uniquely corresponding to a clock buffer and the node set V covers all of the clock buffers. There is an edge between every pair of nodes, i.e., this is a complete graph. Then, a bi-partitioning [7] is performed on G to partition V into two disjoint subsets V_+ and V_- such that $V = V_+ \cup V_-$ and $||V_+| - |V_-|| \leq 1$. The subsets V_+ and V_- correspond to positive and negative polarities, respectively.

If two clock buffers are very close to each other, we prefer to separate them into different subsets (polarities). In a typical graph bi-partitioning [7], two nodes with a small weight edge in-between are more likely to be separated into two subsets. Thus, we let the weight of and edge (i, j) be d_{ij} which is the distance between node i and j. Since a typical bi-partitioning algorithm minimizes the total weight of edges in the cut, an edge with small weight (or distance) has a large chance to be in the cut and its two end nodes are separated in different subsets.

4.2 2-coloring on Minimum Spanning Tree

This is a very simple yet effective technique. First, a minimum spanning tree is generated for the nodes representing clock buffers. Again, each edge weight is defined as the distance between its two incident nodes. Then, a 2-coloring procedure is applied on the minimum spanning tree. In 2-coloring, two end nodes of an edge are always assigned with different colors (or polarities). For a tree, there is always a feasible solution for 2-coloring and it can be found easily. Each color corresponds to a polarity. Since the minimum spanning tree algorithm chooses short edges, two nodes close to each other have opposite polarities.

4.3 Recursive Min-matching

A graph $G = (V, E)$ same as that in Section 4.1 is constructed. Performing min-matching (minimum weighted matching) on this graph results in about $|V|/2$ matched node pairs. In a min-matching, the total weight of the edges between matched nodes is minimized among all possible matchings. Then, we force the two nodes (clock buffers) in the same pair to have opposite polarities. Since the min-matching algorithm normally selects pairs corresponding to small edge weight, the min-matching based polarity assignment tends to let two nearby buffers have opposite polarities.

However, requiring opposite polarities is not a complete assignment for a pair of buffers. For example, for a pair of clock buffers (a, a'), we can either let a be positive and a' be negative (denoted as (a_+, a'_-)), or let a be negative and a' be positive (a_-, a'_+). Both of the polarity permutations satisfy the constraint of being opposite. We denote the former as positive permutation $(a_+, a'_-)_+$ and the latter as negative permutation $(a_-, a'_+)_-$.

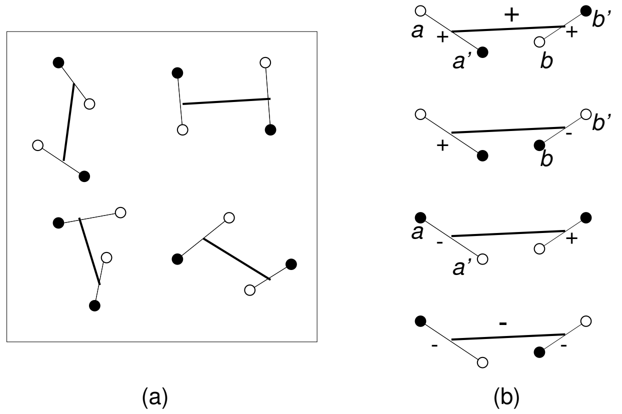

Figure 4: Recursive min-matching.

The selection of polarity permutation is decided by performing another iteration of min-matching on the node pairs obtained in the first min-matching. In this iteration, the nodes of the graph is composed by the centroids of the node pairs matched in the previous iteration. Each edge weight is defined as the distance between corresponding centroids. If two node pairs $((a, a'), (b, b'))$ is selected to be matched in this iteration of min-matching, we have four different polarity permutations: (1) $((a_+, a'_-)_+, (b_+, b'_-)_+)$, (2) $((a_+, a'_-)_+, (b_-, b'_+)_-)$, (3) $((a_-, a'_+)_-, (b_+, b'_-)_+)$ and (4) $((a_-, a'_+)_-, (b_-, b'_+)_-)$. The notations for these permutations can be abbreviated as $++, +-, -+$ and $--$. These four cases are illustrated in Figure 4(b). It can be seen that $++$ and $--$ have no difference to the four clock buffers themselves. Similarly, $+-$ and $-+$ are equivalent to each other for the four buffers. But, $++$ and $--$ are different from $+-$ and $-+$. For the example of Figure 4(b), it is obvious that permutation $++$ (at top) and $--$ (at bottom) are better than permutation $+-$ and $-+$ (in middle). Therefore, we choose $++$ and $--$ it can be fully denoted by $((a_+, a'_-)_+, (b_+, b'_-)_+)_+$ and $((a_-, a'_+)_-, (b_-, b'_+)_-)_-$, respectively. The former is called positive permutation and the later is negative permutation. Now we have multiple node groups, each of which contains four nodes. The min-matching and polarity permutation selection can be repeated recursively on them till there is a single group containing all nodes.

5. BUFFER TYPE SELECTION AND POST PROCESSING

After buffer signal polarity assignment, we need to choose either inverting or non-inverting type for each clock buffer. This procedure is straightforward. If a buffer has the same polarity as its parent buffer, it should use non-inverting type. Otherwise, an inverting type is applied.

In traditional clock tree designs, people prefer to use the same number of buffers on each source-sink path and use the same buffer type at each level [13]. Such design can make clock skew robust to inter-die process variations. However, our buffer polarity assignment may result in different buffer types at a specific level. Therefore, we try to match the buffer types without affecting signal polarity in a post processing. After the buffer type matching, buffer sizing is performed to restore the original clock skew. Both the buffer type matching and buffer sizing are focused on flip-flops which are sequentially adjacent[1], because the fundamental timing constraints - setup time and hold constraints, are mainly for sequentially adjacent flip-flops.

Since the original clock skew is changed due to the buffer type change in the polarity assignment, we run a clock skew tuning procedure after the buffer type matching to restore the original clock skew. This tuning procedure is same as [8] where the sizes of dummy capacitors are tuned toward desired clock skew.

6. EXPERIMENTAL RESULTS

The proposed procedure for power noise reduction was implemented in C on a Windows machine using 1GB RAM. The initial clock trees were obtained using the ISCAS89 benchmark circuits. The circuits were synthesized and technology mapped using SIS [9]. The positions of the clock sinks were obtained using an academic placer mPL [10]. The clock tree is then constructed using DME [12] and the clock buffers are placed similar as [13]. Other technology constants were obtained from [11] (using 180nm model cards) and V_{dd} was set to 2.5V. Table (2) details the characteristics of various test cases.

Table 2: Testcases.

Case	# Sinks	# Buffers
S9234	135	20
S5378	164	25
S13207	503	77
S38584	1426	235
S35932	1728	286

To measure the effectiveness of our technique, we perform simulations to determine the peak current, power supply noise, rise/fall delay variation, power consumption, total capacitance and global skew. For each parameters, we insert several sampling points in the circuit to measure the value during SPICE transient simulation. We record the worst case at each sampling point. For delay variation, we set some logic gates that share the power grid with clock buffers, and measure the delay of these logic gates. We use an inverter of size 5/10 driving a load capacitance of 40fF as our logic gate.

In the data Tables (3,4,5), we report the average and the maximum results among these worst case values from different sampling points. Each table consists of set of 5 columns. The first set of columns presents the results of the base case. The second set presents the previous work [5] . This is followed by a set of three columns that present the the three proposed algorithms i.e. Partition, MST and Matching. The final row in each table reports the normalized average of each parameter.

We post process the clock tree obtained (after assigning different polarities) to tune the skew by techniques suggested in [8]. By doing that, we bring the skew to be less than the required skew bound for all the test cases. The skew bound was set to 10$psec$ for the first two test cases, 25$psec$ for the third test case and 50$psec$ for the final two. The average values of CPU run time for the three proposed algorithms are below 1 sec. Since the run-time is negligible, our technique offers the flexibility of trying all three approaches and picking the one that offers the best results. The following observations could be drawn from the results:

- Our techniques clearly dominate the method suggested in [5] in terms of peak current, power supply noise, rise/fall delay variation, power consumption and total capacitance.
- The reduction in peak current is significant - 38%, 44% and 42% respectively for partition, MST and minimum matching algorithms. In fact, in few cases it could lead up to more than 50% peak current reduction. Such high reductions in peak current have a direct positive impact on circuit reliability.
- The power supply noise, rise delay variations and fall delay variations come down by 44-50%, 50-54% and 40-45% respectively for the three different algorithms.
- The impact of our algorithm for delay variation reduction is higher for the bigger clock networks. For one benchmark (s38584), MST algorithm leads to 75% reduction in the rise delay variation. This trend is encouraging as it indicates that our algorithm scales favorably for bigger nets.
- The total power consumption reduces by 24-30% for the three different algorithms.

Table 6: Results for Total Power Consumption(mW)

Case	Base	[5]	Partition	MST	Matching
s5378	18.7	18.1	13.3	14.3	14.6
s9234	16.1	15.1	13.8	11.1	9.8
s13207	58.6	52.5	46.4	44.3	41.3
s35932	235.0	214.8	171.3	177.2	181.2
s38584	160.8	175.7	131.0	127.0	119.8
Nor Ave.	1.00	0.94	0.76	0.73	0.70

7. CONCLUSIONS

In this work, we propose techniques to reduce the clock network induced power supply noise by assigning different polarities to the clock buffers in an existing clock tree. We detail three different algorithms for the same problem. Experimental results indicate significant reduction in peak current, power supply noise and delay variations. Such reductions in peak current and delay variations lead to more reliable clock networks. As V_{dd} goes lower (than the 2.5V used in our experiments), we expect to see greater delay variations [3] in tree with no polarity assignment. In such a scenario, our work will have a bigger impact.

8. ACKNOWLEDGMENT

The authors would like to thank Suhail Ahmed at Freescale Semiconductor for some helpful discussions.

[1] A pair of flip-flops are sequentially adjacent if there is a pure combinational logic path in-between.

Table 3: Results for Peak Current(mA)

Case	Base Case		Previous Work [5]		Partition		MST		Matching	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
s5378	47.8	70.0	46.3	67.9	31.8	65.4	31.7	65.4	30.2	65.4
s9234	54.6	81.6	42.0	63.4	33.3	63.4	22.0	63.4	23.9	63.4
s13207	133.5	179.6	114.7	166.8	76.4	105.2	68.9	89.2	68.1	90.6
s35932	168.0	228.0	150.0	192.6	114.6	148.6	115.5	160.2	108.2	141.4
s38584	185.5	247.7	135.6	198.8	113.9	146.5	98.6	144.7	125.4	163.4
Nor Ave.	1.00	1.00	0.85	0.86	0.62	0.70	0.56	0.69	0.58	0.69

Table 4: Results for Power Noise(V)

Case	Base Case		Previous Work [5]		Partition		MST		Matching	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
s5378	0.088	0.125	0.075	0.120	0.047	0.071	0.047	0.068	0.045	0.075
s9234	0.100	0.150	0.051	0.083	0.040	0.060	0.023	0.051	0.260	0.053
s13207	0.271	0.353	0.230	0.320	0.150	0.210	0.130	0.181	0.130	0.181
s35932	0.680	0.850	0.581	0.754	0.480	0.602	0.470	0.580	0.450	0.580
s38584	0.645	0.792	0.501	0.623	0.401	0.520	0.390	0.501	0.450	0.550
Nor Ave.	1.00	1.00	0.77	0.82	0.56	0.58	0.50	0.54	0.52	0.56

Table 5: Results for Delay Variation(ps)

Case	Base Case		Previous Work [5]		Partition		MST		Matching	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
s5378	0.65	0.72	0.62	0.70	0.48	0.54	0.47	0.53	0.47	0.54
s9234	0.33	0.36	0.32	0.35	0.25	0.32	0.20	0.24	0.23	0.27
s13207	1.21	1.44	1.11	1.40	0.86	1.11	0.80	1.06	0.79	1.04
s35932	9.81	14.43	7.61	12.80	4.07	10.05	4.83	10.62	3.72	9.87
s38584	8.29	12.77	4.85	9.95	3.12	9.25	2.49	8.07	3.19	9.50
Nor Ave.	1.00	1.00	0.84	0.91	0.60	0.77	0.55	0.70	0.56	0.70

9. REFERENCES

[1] R. Saleh, S. Z. Hussain, S. Rochel, and D. Overhauser. Clock skew verification in the presence of IR-drop in the power distribution network. *IEEE TCAD*, 19(6):635–644, June 2000.

[2] L. H. Chen, M. Marek-Sadowska, and F. Brewer. Buffer delay change in the presence of power and ground noise. *IEEE TVLSI*, 11(3):461–473, June 2003.

[3] S. S. Sapatnekar and H. Su. Analysis and optimization of power grids. *IEEE Design and Test of Computers*, 20(3):7–15, May-June 2003.

[4] A. Vittal, H. Ha, F. Brewer, and M. Marek-Sadowska. Clock skew optimization for ground bounce control. In *Proceedings of the IEEE/ACM ICCAD* , pages 395–399, 1996.

[5] Y.-T. Nieh, S.-H. Huang, and S.-Y. Hsu. Minimizing peak current via opposite-phase clock tree. In *Proceedings of the ACM/IEEE DAC* , pages 182–185, 2005.

[6] K. Wang, Y. Ran, H. Jiang, and M. Marek-Sadowska. General skew constrained clock network sizing based on sequential linear programming. *IEEE TCAD*, 24(5):773–782, May 2005.

[7] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: a survey. *Integration: the VLSI Journal*, 19(1-2):1–81, 1995.

[8] G. Venkataraman, N. Jayakumar, J. Hu, P. Li, S. Khatri, A. Rajaram, P. McGuinness and C. Alpert. *Practical Techniques to Reduce Skew and its Variations in Buffered Clock Networks.* IEEE/ACM ICCAD, pages 591–595, 2005.

[9] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. SIS: a system for sequential circuit synthesis. Memorandum no. M92/41, ERL, ,UCB May 1992.

[10] CPMO-constrained placement by multilevel optimization. http://ballade.cs.ucla.edu/cpmo/. Computer Science Department, UCLA.

[11] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation. In *Proceedings of the IEEE CICC*, pages 201–204, 2000.

[12] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese and A. B. Kahng, Zero skew clock routing with minimum wirelength. *IEEE Transactions on CAS-39(11)*,pages 799–814, November, 1992.

[13] S. Pullela, N. Menezes, J. Omar and L. T. Pillage. Skew and delay optimization for reliable buffered clock trees. In *Proceedings of the IEEE/ACM ICCAD*, pages 556-562, 1993.