Clock Gating Optimization with Delay-Matching

Shih-Jung Hsu
Computer Science and Engineering
Yuan Ze University
Chung-Li, Taiwan

Rung-Bin Lin
Computer Science and Engineering
Yuan Ze University
Chung-Li, Taiwan

***Abstract*—Clock gating is an effective method of reducing power dissipation of a high-performance circuit. However, deployment of gated cells increases the difficulty of optimizing a clock tree. In this paper, we propose a delay-matching approach to addressing this problem. Delay-matching uses gated cells whose timing characteristics are similar to that of their clock buffer (inverter) counterparts. It attains better slew and much smaller latency with comparable clock skew and less area when compared to type-matching. The skew of a delay-matching gated tree, just like the one generated by type-matching, is insensitive to process and operating corner variations. Besides, delay-matching ECO of a gated tree excels in preserving the original timing characteristics of the gated tree.**

Keywords- Clock gating; low power design; clock tree

I. INTRODUCTION

Clock gating is an effective method of reducing power dissipation of a high-performance circuit [1-14]. As shown in Fig. 1, En1 and En2 input to the AND gates can be employed to prevent clock signals from reaching the downstream flip-flops and hence reduce the switching activity of flip-flops. However, deployment of gated cells such as AND gates increases the difficulty of synthesizing a low-skew gated tree. Arguing that different types of logic gates employed in a gated clock tree are detrimental to clock balancing based on load-matching mechanism, the work in [14] proposes using type-matching gates to grow a clock tree based on load-matching mechanism. Type-matching prescribes that all the logic gates on the same level must be of the same type. The logic gates on the same level can be either all inverters, buffers, NAND, AND, NOR, or OR gates. Although type-matching approach can result in lower clock skew, it may incur excessive clock latency and bad slew at buffers and clock sinks due to having only a limited number of gated cell types in a typical standard cell library. It may also cause a design to use more cell areas.

In this paper, we propose delay-matching concept to address some problems about type-matching. Delay-matching is achieved using gated cells whose timing characteristics are similar to that of their clock buffer (inverter) counterparts. Hence, a traditional clock tree synthesizer can be employed to obtain a delay-balanced clock tree just treating a gated cell as a clock buffer (inverter). Delay-matching concept can thus be easily implemented and integrated into an industrial design flow. To facilitate such a task, we also propose an approach to designing delay-matching cells and integrating them into a commercial standard cell library. We apply delay-matching to three occasions of clock gating.

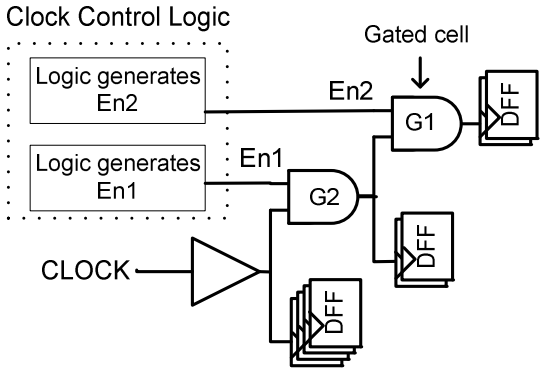

Figure 1. A gated clock design.

- Optimizing a homogeneous gated tree where the clock paths from the root to all the sinks are of the same depth. Experimental results based on UMC 90nm process technology show that delay-matching attains better clock slew and much smaller clock latency with comparable clock skew at the typical, worst, and best corners when compared with type-matching approach. Delay-matching achieves all of these using less area than type-matching does. Besides, the skew of a delay-matching gated tree, just like the one generated by type-matching, is insensitive to process and operating corner variations. Moreover, the slews of a delay-matching gated tree are less sensitive to process and operating corner variations than that of a type-matching gated tree.
- Optimizing a non-homogeneous gated clock tree. Our results show that delay-matching obtains smaller latency and better slew but incurs larger area and skew.
- Performing ECO (Engineering Change Order) of a gated tree. Our results show that delay-matching ECO excels in preserving the original timing characteristics of a gated tree.

Note that delay-matching is a general concept. Its applications are not limited to the occasions presented in this work. One of its disadvantages is that we need to re-engineer a cell library to include delay-matching cells.

The rest of this article is organized as follows. Section II presents some basics and problems about type-matching clock gating. Section III describes our delay-matching concept and how we design delay-matching cells. Section IV presents how delay-matching is employed to optimize a gated tree and shows some experimental results. The last section draws conclusions.

978-3-9810801-7-9/DATE11/©2011 EDAA.

II. TYPE-MATCHING CLOCK GATING

Type-matching concept is proposed in [14] to reduce clock skew in a gated clock tree. Its main motivation is to address the problem of large clock skew caused by different timing characteristics of gated cells and buffers. Without showing the control logic, Fig. 2 presents a gated clock tree based on type-matching concept for the clock design in Fig. 1. As one can see, the type-matching tree is augmented with four gates T1, T2, T3, and T4. The gates on each level of the clock tree must be of the same type and the same driving capability. For example, G1, T2, T3, and T4 must be of the same type and the same driving capability. A traditional clock tree synthesizer can be employed to minimize the skew of a type-matching tree. As shown in Fig. 3, a non-type-matching tree may use different types of cells on a level. It is harder to minimize the skew of such a tree due to a discrepancy between AND's and buffer's timing characteristics. Note that if a clock tree has inverters, NAND gates will be employed to grow a type-matching tree. One problem pointed out in [14] is that if a gated clock design employs both AND and OR gates on the same level, NAND gates should be used to implement both AND and OR gates. Fig. 4 shows such an example for the case if T1 in Fig. 2 is replaced by an OR gate with a control input En3. In Fig. 4, T5 and T6 implement an AND gate and T7 and T8 implement an OR gate to form a type-matching gated tree. However, not mentioned in [14] is that doing so may increase clock latency, power consumption, and chip area.

Type-matching has yet another problem if a typical cell library is used. A typical cell library consists of gated cells only up to 5X or 6X driving capability. To achieve good slew, the fanout driven by a gated cell must be kept small. This will increase the size of a clock tree considerably. Moreover, the clock tree tends to have more buffering levels so that it is less likely to have a small latency.

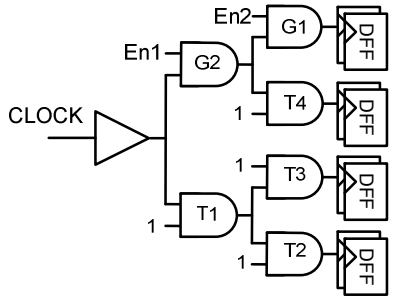

Figure 2. Clock gating with type-matching cells.

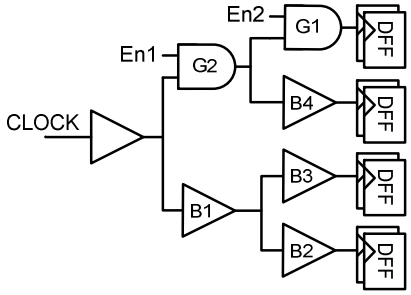

Figure 3. Clock gating without type-matching, or clock gating with delay-matching if delays of G1, B2, B3, and B4 are the same and delays of G2 and B1 are also the same.

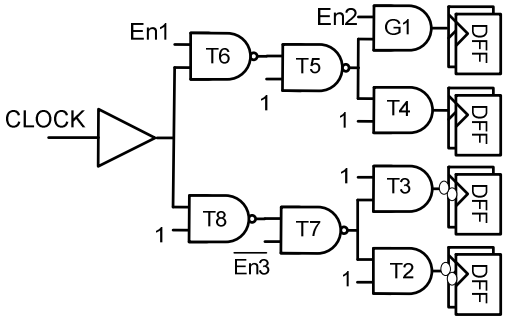

Figure 4. Type-matching gated tree for the clock design in Fig. 2 with T1 being replaced by an OR gate with control input En3.

A. Type-Matching with High-Drive Gated Cells

As discussed above, the gated cells in a typical standard cell library have only a limited number of driving classes. Hence, in our work we consider extending a typical cell library to include large-drive gated cells. These large-drive gated cells can be used to replace large-drive clock buffers and inverters. However, they are not delay-matching. They are designed only to match the driving capabilities of clock buffers and inverters. High-drive type-matching can improve clock latency and slews at buffers and sinks. However, it still causes a design to use a large total cell area.

III. DELAY-MATCHING CLOCK GATING

A. Delay-Matching Concept

The basic idea of delay-matching is to employ a set of gated cells which have the same timing characteristics as their clock buffer (inverter) counterparts. Take the clock tree in Fig. 3 for example. Let B2, B3, and B4 be instances instantiated from the same clock buffer, said CKBUF4X, and B1 be an instance instantiated from a buffer, said CKBUF8X. Let G1 be an instance of CKAND4X and G2 be an instance of CKAND8X. If CKAND4X is a delay-matching cell for CKBUF4X and CKAND8X is a delay-matching cell for CKBUF8X, Fig. 3 gives a delay-matching gated clock tree. Clearly, if we count only the gate and buffer delays on the paths to sinks, this clock tree is supposedly delay-balanced. Its slew and latency would be similar to that of the clock tree without clock gating.

To realize a delay-matching tree, a gated cell should have the following properties.

- Taking AND gate for example, if there is a clock buffer with *k*X driving capability in a standard cell library, there will be a two-input AND gate with *k*X driving capability. This AND gate is called CKAND*k*X and its clock buffer counterpart is called CKBUF*k*X.
- The clock input capacitance of CKAND*k*X is similar to that of CKBUF*k*X.
- The rise time, fall time, rise delay, and fall delay of CKAND*k*X are similar to that of CKBUF*k*X.

The first property is to provide a variety of gated ANDs, some of them with large driving capability. The second and third properties make our gated ANDs look like clock buffers so that load balancing and thus delay balancing during growing a clock tree can be achieved with ease.

Similarly, CKORkX for its CKBUFkX counterpart and CKNANDkX and CKNORkX for its CKINVkX counterpart should also possess these properties.

B. Designing of Delay-Matching Cells

To design a delay-matching cell whose timing characteristics are similar to its counterpart, we must make the cell satisfy the last two properties presented above. Taking CKNAND1X as shown in Fig. 5 for example, we would like to make CKNAND1X similar to CKINV1X. Given that CKINV1X has a PMOS with channel width $w = 1860nm$ and an NMOS with channel width $w = 545nm$ and their timing characteristics, we would like to determine the widths of P1, P2, N1, and N2 of CKNAND1X. Since we assume that clock control signal will arrive at input En much earlier than clock signal, we should make P2 as small as possible to reduce cell area. In our case, the width of P2 is set to 360nm. As for the widths of P1, N1, and N2, we perform HSPICE simulation to determine their values. Our approach has the following three steps.

- For each pair of input slew (totally seven slews) at Clk and an output load (totally seven loads) at Y, HSPICE simulations are performed to determine the widths of P1, N1, and N2 such that the timing characteristics of CKNAND1X are similar to that of CKINV1X. We do this for all the 49 pairs of input slews and output loads. We hence obtain 49 width values of P1, N1, and N2, respectively. Note that HSPICE allows us to perform such simulation easily. This requires us to guess an initial width and a possible width range for each transistor at each simulation.
- For each transistor, we find its average, maximum, and minimum widths from the 49 data sets. We use the average width as the width of the transistor and perform timing characterization of CKNAND1X with such width. We then find out the timing discrepancy TD, a sum of the absolute timing differences between CKNAND1X and CKBUF1X for all the 49 pairs.
- With initial width set equal to the average width and its possible width range defined by the minimum and maximum widths obtained from the previous iteration, we repeatedly perform the above two steps until TD cannot be further reduced. The transistor widths are then determined by the iteration with the smallest TD.

In the above procedure, we consider all the 49 pairs for determining transistor's width. However, we find that with so-obtained transistor sizes, a gated cell tends to have large timing discrepancies. Based on our observation, the transistor widths for the pairs with smaller slews and smaller loads tend to be much larger than the others. Hence, we employ only the 20 pairs formed by larger input slews and output loads to determine transistor widths. The timing data presented hereafter are obtained using the transistor widths determined in such a way. Note that the above task for determining transistor widths of a delay-matching cell can be automated easily and efficiently.

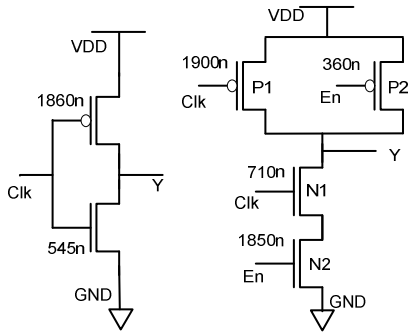

Figure 5. Transistor sizing of CKNAND1X (right) for matching CKINV1X

How similar are our gated cells to their counterparts in terms of timing behavior? TABLE I shows a comparison of clock input capacitance of gated cells with that of their buffer (inverter) counterparts. As one can see, the input capacitance of CKNANDkX and CKANDkX is quite close to that of CKINVkX and CKBUFkX, respectively. This is a direct consequence of forcing the equivalence of timing characteristics between gated cells and their counterparts. TABLE II shows the average and maximum timing discrepancies between CKINVkX and CKNANDkX. The second to fifth columns show the discrepancies for the case of counting the data of all 49 pairs and the sixth to 9th columns show the discrepancies for the case of counting only the data of 20 pairs with larger slews and loads. As one can see, timing discrepancy is on average small except for some cases. Especially, the discrepancy in rise delay (RD) between CKINVkX and CKNANDkX is large. The large discrepancy is mainly due to some data points which have small denominators in calculating discrepancies so that the average discrepancy is widened by these data points considerably. Similar results are observed for CKBUFkX and CKANDkX.

It is worthwhile to mention that our delay-matching cells CKNANDkX has similar timing characteristics as that of CKNORkX because CKNANDkX and CKNORkX are both designed to have similar timing characteristics as that of CKINVkX. By the same reason, both CKANDkX and CKORkX have similar timing characteristics. For example, the timing discrepancies between our CKAND1X and CKOR1X on the 49 pairs of slew and load are on average 0.83% for rise delay, 2.68% for rise time, 0.85% for fall delay, and 2.89 for fall time. The maximum discrepancies are 9.13%, 19.81%, 7.39%, and 14.89%, respectively.

TABLE I. INPUT CAPACITANCE OF GATED CELLS AND THEIR COUNTERPARTS (PF).

	CKINV	CKNAND	CKNOR	CKBUF	CKAND	CKOR
1X	0.00391	0.00395	0.00455	0.0039	0.00418	0.00478
2X	0.00704	0.00735	0.00841	0.0039	0.00442	0.00479
3X	0.01015	0.01067	0.01234	0.00703	0.00738	0.008
4X	0.01337	0.0138	0.01625	0.00703	0.0076	0.00837
6X	0.01963	0.01988	0.0237	0.01015	0.01073	0.01211
8X	0.02588	0.02694	0.03095	0.01337	0.01392	0.01567
12X	0.03849	0.03983	0.0465	0.01962	0.02052	0.02198
16X	0.05113	0.05193	0.06348	0.02588	0.02748	0.02867
20X	0.06374	0.0668	0.07749	0.03223	0.03373	0.03641

TABLE II. TIMING DISCREPANCIES (AVERAGE%/MAX%) BETWEEN CKINVKX AND CKNANDKX. (RD: RISE DELAY, RT: RISE TIME, FD: FALL DELAY, FT: FALL TIME).

	49 pairs				20 pairs			
	RD	**RT**	**FD**	**FT**	**RD**	**RT**	**FD**	**FT**
1X	6/51	6/30	3/11	4/21	4/7	3/10	2/6	4/12
2X	90/2707	5/30	3/20	4/18	4/24	4/13	2/4	3/16
3X	39/661	6/40	5/17	4/23	13/185	5/40	4/10	5/10
4X	81/2741	8/104	4/18	3/17	6/55	3/8	3/8	3/15
6X	28/339	6/42	4/17	4/19	19/271	3/11	3/6	5/19
8X	33/716	6/41	3/8	3/22	43/716	5/29	2/5	5/22
12X	42/834	7/51	3/14	3/23	8/90	5/18	3/14	5/23
16X	80/2559	7/41	4/11	3/22	136/2559	4/18	2/6	4/22
20X	42/509	5/26	3/15	3/22	15/204	4/12	2/15	4/22

The good thing for the similarity of timing characteristics between CKANDkX and CKORkX is that we can easily address the problem caused by having both AND and OR gates on the same level of a clock tree. Let us redraw the clock tree in Fig. 4 by replacing T5 and T6 by an AND gate G2, T7 and T8 by an OR gate G3, and the type-matching AND gates by buffers, as shown in Fig. 6. Now, we obtain a gated clock tree with different gated cells, i.e., G2 and G3, on the same level. Because G2 and G3 have similar timing characteristics and G1, B1, B2, and B3 also have similar timing characteristics due to delay-matching, we obtain a delay-balanced gated clock tree.

Once the transistor sizes of a delay-matching cell are determined, layout design then follows. The cell height and power/ground rail widths of a delay-matching cell are made the same as those of the other cells in an industrial standard cell library based on UMC 90um technology. Figure 7 shows layouts of some delay-matching cells. The delay-matching cells are characterized and added into the underlying standard cell library. Such a library is called delay-matching library.

C. Discussion

There are still some problems needed to be addressed for delay-matching and type-matching approaches. As shown in Fig. 8, the second level of the clock tree employs an inverter, a buffer, and an OR gate. With type-matching, the OR gate and buffer each can be replaced by two serially connected NAND gates. However, this can not be done for the inverter. Hence, the clock tree in Fig. 8 can not be transformed into a type-matching gated tree. Similarly, with delay-matching, the OR gate and buffer have similar timing characteristics, but the inverter and buffer don't. Hence, the clock tree in Fig. 8 can not be transformed into a delay-matching gated tree, either. However, if we can design an inverter whose timing characteristics are made similar to that of a buffer, we can still transform the tree into a delay-matching gated tree. Based on our experiments, we can achieve on average 10%, 20%, 17%, and 23% timing discrepancy between buffers and inverters for rise delay, rise time, fall delay, and fall time, respectively. Here, we architect a three-stage inverter so that its timing characteristics can be made as closer to that of its buffer counterpart as possible.

IV. SYNTHESIS OF DELAY-MATCHING GATED CLOCK TREE

Advantageously, we do not need any special algorithm for synthesizing a delay-matching gated clock tree. A gated clock tree synthesis algorithm like that presented in [8-11] should be sufficient. For simplicity, we assume that we have a clock control logic unit synthesized based on flip-flops activity

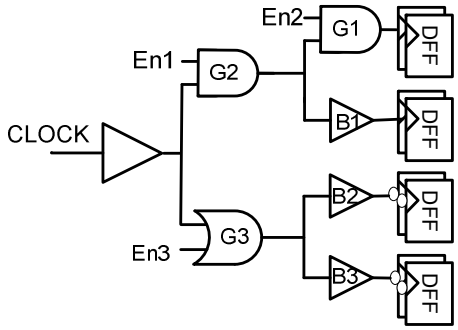

Figure 6. Delay-matching with different types of gated cells

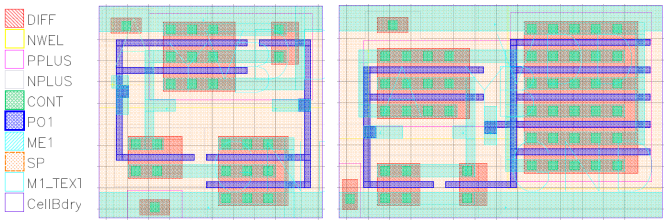

Figure 7. Layouts of CKNAND1X (left) and CKNOR1X(right).

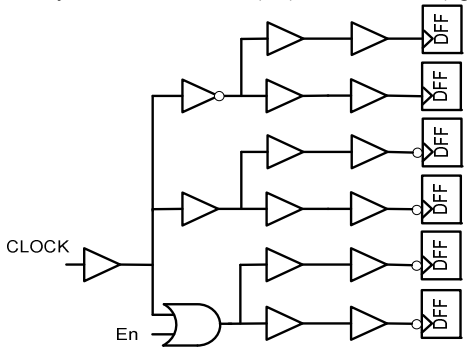

Figure 8. A gated clock tree with opposite triggering phases.

patterns for generating clock gating signals. Here, we apply delay-matching to three occasions of clock gating.

A. Delay-matching on a Homogeneous Gated Tree

A homogeneous gated tree is a clock tree where each path from the root to a sink traverses the same number of gates, including gated cells, clock buffers, and clock inverters, as shown in Fig. 3. For delay-matching, we replace the gated cells like G1 and G2 in Fig. 3 with our delay-matching cells. Once we have a delay-matching gated tree, we perform clock tree placement and routing to complete the clock tree design using some commercial tools.

For the purpose of comparisons, we implement the following clock gating methods.

NG: non-gating.

NML: Normal clock gating using an industrial standard cell library based on UMC 90nm process technology. It is neither delay-matching nor type-matching.

DM: Delay-matching clock gating.

SmallT: Type-matching using small-drive gated cells.

BigT: Type-matching using also large-drive gated cells.

- Synthesis of a DM gated tree

We first obtain a design without clock gating using Synopsys Design Compiler. We then place the design and perform clock tree synthesis with skew and slew constraints using Cadence SOC Encounter. To this point, we obtain a

buffered and placed clock tree without clock gating. We randomly select a certain percentage of clock buffers (inverters) in the clock tree and change them into corresponding gated cells. For simplicity, we wire together all the clock control signals on the same level. At this moment, we should obtain a clock tree like the one shown in Fig. 3. We place gated cells at the locations where their individual counterparts are located originally. Because gated cells are larger than their buffer (inverters) counterparts, we perform cell legalization using Cadence SOC Encounter to remove cell overlaps. We then use Encounter to complete clock tree and signal routing and obtain clock skew, latency, and slew at buffers and sinks.

- Synthesis of a BigT gated tree

We use a cell library augmented with large-drive gated cells for synthesizing a BigT gated tree. For simplicity, rather than using the type-matching algorithm presented in [14], we adopt an approach similar to our delay-matching one. The way of obtaining a type-matching gated tree using BigT is similar to that of delay-matching except that, after obtaining a clock tree like the one shown in Fig. 3, we replace buffers (inverters) with type-matching AND (NAND) gates to obtain a type-matching tree like the one shown in Fig. 2.

- Synthesis of a SamllT gated tree

We use a cell library that only has clock buffers and inverters of driving capabilities 1X, 2X, 3X, 4X, and 6X for clock tree synthesis. Similar to delay-matching approach, we also randomly replace a certain percentage of buffers and inverters with gated cells and obtain a gated clock tree like the one shown in Fig. 3. We then perform type-matching synthesis to obtain a gated clock tree like the one shown in Fig. 2.

We evaluate the above clock gating approaches using three large ISCAS89 benchmark circuits (s35932, s38417, s38584) and eight large ITC99 benchmark circuits (b14, b15, b17~b22). These benchmarks have a few to 100 thousand equivalent NAND gates. Remember that we randomly replace a certain percentage of clock buffers (inverters) with gated cells. Here, we try 15%, 30%, and 50%. We consider typical, best, and worst corners. The typical/best/worst corner uses a typical/fast/worst SPICE model at 25/-40/125 degree C and 100%/90%/110% VDD supply voltage.

TABLE III shows the average values of various performance indices obtained by the above clock-gating approaches. At typical corner, NML is obviously not viable. DM (delay-matching) has its latency, skew, and slew close to that of NG (non-gating). Note that maintaining clock latency is important for re-spinning a design to implement clock gating. Delay-matching achieves substantial power saving with respect to NG at the expense of more cell areas. Power data include switching and leakage powers of clock tree but do not include switching and leakage powers of flip-flops. Note that a more significant power saving should be obtained from eliminating unnecessary switching of flip-flops and the logic driven by flip-flops. To further verify the data obtained by Encounter, we also use Encounter to generate a SPICE netlist of a clock tree and use HSPICE to simulate the clock tree to obtain power, latency, skew, and slews at buffers and sinks. These data are presented in the right-most five columns of TABLE III.

Compared to type-matching approach SmallT, DM obtains a much better latency and better slews at buffers and sinks with a slightly worse skew. Also, DM uses smaller cell area and incurs smaller wire length. The reason why SmallT uses more cell area and larger wire length is because SmallT creates a large gated tree (due to smaller fanout driven by per driver) in order to satisfy the slew constraint at buffers and sinks. This also causes type-matching to create a deep gated tree so that its latency is significantly larger than that of a delay-matching gated tree.

We observe from TABLE III that using BigT improves latency and slews even though the latency and slews are still worse than that of DM. However, BigT incurs larger skew and uses considerably more cell area than DM. Since the source/drain capacitance of a large-drive gated cell is larger than that of its buffer (inverter) counterpart, the latency of a BigT gated tree is still larger than that of a delay-matching gated tree.

TABLE III also shows some results at the best and worst corners. As one can see, the skews of both delay-matching gated tree and type-matching gated tree are not sensitive to corner variations. However, it seems that the slews of a delay-matching gated tree are less sensitive to corner variations than that of a type-matching gated tree. The reason for this may be that a delay-matching gated tree provides a larger drive than a SmallT gated tree does. Or each of its buffers (inverters) drives a smaller load than that of a BigT gated tree.

TABLE III. RESULTS FOR HOMOGENEOUS GATED TREE SYNTHESES USING DIFFERENT GATING METHODS (BS: WORST SLEW AT BUFFERS (INVERTERS); SS: WORST SLEW AT CLOCK SINKS; CTW: TOTAL WIRE LENGTH OF CLOCK ROUTING; CTA: TOTAL CELL AREA OF A GATED CLOCK TREE).

	Gating methods	Latency (ps)	Skew (ps)	BS (ps)	SS (ps)	CTW (um)	CTA (um^2)	Power (mW)	HSPICE				
									Power	Latency	Skew	BS	SS
Typical corner	**NG**	146.32	15.97	19.40	18.95	12251	1264	2.31	2.17	168.06	17.37	24.60	22.25
	NML	206.78	80.78	31.63	32.46	12271	1267	1.70	1.67	228.02	82.42	41.89	35.71
	DM	150.10	19.84	21.45	20.18	12458	1673	1.86	1.77	173.14	22.16	27.64	24.20
	SmallT	280.47	14.45	38.10	23.00	13387	2558	1.90	1.96	295.35	15.88	42.23	25.83
	BigT	189.48	22.01	27.02	25.09	12517	2349	2.03	2.09	209.42	24.64	37.23	28.05
Best corner	**NG**	112.12	16.42	16.33	15.83	12251	1264	2.87	3.06	140.83	18.26	24.12	20.19
	NML	152.95	59.76	24.52	25.59	12271	1267	2.13	2.37	185.52	66.93	36.51	30.70
	DM	116.63	21.03	18.85	17.75	12458	1673	2.39	2.50	146.35	23.46	27.79	22.30
	SmallT	202.27	13.58	27.90	18.44	13387	2558	2.58	3.05	233.46	15.37	37.39	22.26
	BigT	146.07	23.84	22.39	21.29	12517	2349	2.73	3.28	176.86	26.95	35.55	25.96
Worst corner	**NG**	210.94	16.57	24.82	24.11	12251	1264	1.98	1.58	231.83	17.42	29.34	27.24
	NML	310.65	123.56	44.25	45.32	12271	1267	1.46	1.22	324.09	118.03	52.85	46.44
	DM	214.59	19.91	26.44	25.35	12458	1673	1.56	1.27	236.43	22.06	31.25	28.98
	SmallT	440.69	17.49	56.38	33.45	13387	2558	1.60	1.34	438.26	18.99	54.87	36.04
	BigT	276.08	21.73	35.74	33.50	12517	2349	1.83	1.44	288.92	23.91	43.12	35.25

B. Delay-matching on a Non-homogeneous Gated Tree

In this section, we employ a delay-matching library (DM) and a conventional cell library (UMC90) respectively to synthesize a gated clock tree using Synopsys Design Complier. The so-obtained gated tree is usually non-homogeneous, i.e., not every path from the root to a sink traversing the same number of gates. The tree is still yet to be buffered, placed and routed to meet the skew, slew, and latency constraints. We carry out these tasks using SOC Encounter. TABLE IV shows that delay-matching achieves smaller latency and better slew but larger area and skew. These data are averages of the results for all the benchmarks mentioned in Section IV.A.

C. ECO of Gated Clock Tree with Delay-Matching

We can use delay-matching cells to perform ECO for a gated clock tree so that the skew, latency, and slew at buffers and sinks of the tree can be maintained. As shown in Fig. 9, we can replace a CKBUF*k*X B1 with a delay-matching cell CKAND*k*X G1 or vice versa. Since B1 and G1 have similar timing characteristics, the skew, latency, and slews of these two gated trees will not be much different. To validate this argument, we perform ECO for a non-homogeneous gated tree by replacing one to three buffers by their delay-matching cell counterparts. TABLE V shows the ECO results obtained by using our delay-matching library (DM) and an industrial cell library (UMC90) based on UMC 90nm process technology. UMC90 library is neither for delay-matching nor for type-matching. For each library at each corner, there are two rows of data. The first row gives the data before ECO whereas the second row gives the data after ECO. The data in TABLE V are averages of the results for all benchmarks mentioned in Section IV.A. As one can see, performing ECO for a gated clock tree using a delay-matching cell library can greatly maintain the performance characteristics of the original gated tree. Note that it is important to maintain clock latency of a design with functional ECO (non-performance related ECO).

V. CONCLUSIONS

In this paper we present a delay-matching approach to optimizing a gated clock tree for power reduction. We also develop a method for designing delay-matching cells. Compared with type-matching, delay-matching attains better slew and clock latency with comparable clock skew while using much less cell area. Besides, the skew of a delay-matching gated tree, just like the one generated by type-matching, is insensitive to process and operating corner variations. Meanwhile, delay-matching ECO excels in maintaining the original timing characteristics of a gated tree. The disadvantage of delay-matching is that we have to re-engineer a cell library to include delay-matching cells. Fortunately, this task can be automated easily and efficiently. Delay-matching is a general concept so that its applications should not be limited to those presented in this work.

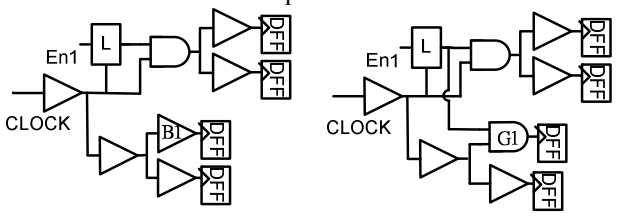

Figure 9. ECO by replacing buffer B1 with gated cell G1 or vice versa.

TABLE IV. RESULTS OBTAINED BY USING DIFFERENT LIBRARIES TO SYNTHESIZE NON-HOMOGENEOUS GATED TREES.

	Library	Latency (ps)	Skew (ps)	BS (ps)	SS (ps)	CTW (um)	CTA (um^2)
Typical corner	**UMC90**	199.6	17.4	26.0	21.5	20993	2203
	DM	176.1	25.7	23.3	19.5	20028	4790
Best corner	**UMC90**	148.7	18.6	21.5	17.0	23541	2459
	DM	143.4	30.2	20.2	17.5	22382	5556
Worst corner	**UMC90**	335.0	28.4	36.0	32.4	23541	2459
	DM	271.7	39.3	30.7	27.4	22382	5556

TABLE V. ECO RESULTS OBTAINED BY USING A DELAY-MATCHING LIBRARY AND AN INDUSTRIAL CELL LIBRARY.

	Library	Latency (ps)	Skew (ps)	BS (ps)	SS (ps)	CTW (um)	CTA (um^2)
Typical corner	**UMC90**	143.3	17.2	21.2	16.7	20993	2203
		154.9	27.3	30.3	17.0	20996	2178
	DM	137.3	27.4	19.6	16.8	20028	4790
		138.3	28.4	19.8	16.8	20045	4816
Best corner	**UMC90**	143.3	17.2	21.2	16.7	20993	2203
		154.9	27.3	30.3	17.0	20996	2178
	DM	137.3	27.4	19.6	16.8	20028	4790
		138.3	28.4	19.8	16.8	20045	4816
Worst corner	**UMC90**	321.5	26.2	36.4	31.7	20993	2203
		359.4	64.1	56.8	32.6	20996	2178
	DM	261.6	34.5	30.1	26.3	20028	4790
		262.0	35.0	30.4	26.4	20045	4816

REFERENCES

[1] L. Benini and G.D. Micheli, "Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines," IEEE Trans. on CAD, Vol. 15, No. 6, pp. 630-643, 1996.

[2] N. Raghavan, V. Akella, and S. Bakshi, "Automatic Insertion of Gated Clocks at Register Transfer Level," International Conference on VLSI Design, pp. 48-54, 1999.

[3] D. Borkovic and K.S. McElvain, "Reducing Clock Skew in Clock Gating Circuits," United States Patent, Patent No. 7082582, 2006.

[4] Q. Wu, M. Pedram, and X. Wu, "Clock Gating and Its Application to Low Power Design of Sequential Circuits," IEEE Trans. on Circuits and Systems – I: Fundamental, Theory, and Applications, Vol. 47, No. 103, pp. 415-420, 2000.

[5] L. Li, K. Choi, S. Park, M. K. Chung, "Novel RT Level Methodology for Low Power by Using Wasting Toggle Rate based Clock Gating," International SoC Design Conference, pp. 484-487, 2009.

[6] E. Arbel, C. Eisner, and O. Rokhlenko, "Resurrecting Infeasible Clock-Gating Functions," DAC, pp. 160-165, 2009.

[7] A. Farrahi, C. Chen, A. Srivastava, G. Tellez, and M. Sarrafzadeh, "Activity Driven Clock Design," IEEE Trans. on CAD, Vol. 20, No. 6, pp. 705-714, 2001.

[8] W. C. Chao and W. K. Mak, "Low-Power Gated and Buffered Clock Network Construction," ACM TODAES, Vol. 13, No. 1, Article 20, January 2008.

[9] D. Garret, M. Stan, and A. Dean, "Challenges in Clock Gating for a Low Power ASIC Methodology," ISLPED, pp. 176-181, 2002.

[10] W. Shen, Y. Cai, X. Hong, and J. Hu, "An Effective Gated Clock Tree Design Based on Activity and Register Aware Placement," IEEE Transactions on VLSI Systems, Vol. 18, No. 12, pp. 1639-1648, 2009.

[11] D. J. Hathaway, "Method for Making Integrated Circuits Having Gated Clock Trees," United States Patent, Patent No. 6536024, 2003.

[12] J. Oh and M. Pedram, "Gated Clock Routing for Low-Power Microprocessor Design," IEEE Trans. on CAD, Vol. 20, No. 6, pp. 715-722, 2001.

[13] C. C. Cheung and K. D. Au, "Clock Gating Cell for Used in a Cell Library," United States Patent, Patent No. 6552572, 2003.

[14] C. M. Chang, S. H. Huang, Y. K. Ho, J. Z. Lin, H. P. Wang, and Y. S. Lu, "Type-Matching Clock Tree for Zero Skew Clock Gating," DAC, 714-719, 2008.