

# Low Power Buffered Clock Tree Synthesis Considering Best Transition Time

## Abstract—

Clocking has been playing a very important role in current VLSI designs. As technology advances, there exist millions or billions of gates on a chip, which makes the clock network design much more challenging since synchronization in chip is one of the primary concerns. However, chips running at higher frequency consume much more power, mostly on clock distribution. Without carefully planning clock network, the chips will suffer from high power dissipation.

In this paper, we develop a methodology which can be applied in buffered clock tree synthesis to achieve low power demands. It is based on analysis on any given buffer library in finding best transition time for low power and customized buffer insertion. The experimental results are encouraging. We have obtained up to 13% power saving and smaller clock skew compared with a previous work based on gate sizing.

## I. INTRODUCTION

Clock designs play an important role in modern VLSI designs. A distributed clock tree structure, as shown in Fig.1, is commonly used in VLSI designs. As technology advances, a chip may have millions of gates with a very complex structure. The synchronization of clocks on a chip is critical to the performance and reliability of the chip. Furthermore, since clock nets have the highest switching activity, it is necessary to reduce the clocking power in high performance designs. Due to low power demand in designing modern chips, clock network synthesis becomes more critical in consuming less power.

Previous works on clock tree construction focused on zero or near zero-skew routing [13] [8]. In addition to zero-skew, more works concentrated on routing clock tree with minimum total wire length and phased delay [7] [2] [4]. Some researches focus on useful-skew to reduce area, wire length and power [12] [15]. Until recently, it is getting more important to reduce the power consumed by the clock network due to higher frequency operation in modern digital systems. For the low power clock, some researches, [10] and [14], have tried to insert buffers to minimize the clock power. [3] proposed to use gate sizing in achieving power optimization. Authors of [5] and [6] exploited similarities in the switching activity of the clocked modules to reduce the number of the gates for clock gating. Tellez et al. in [11] investigated the problem of computing a lower bound on the

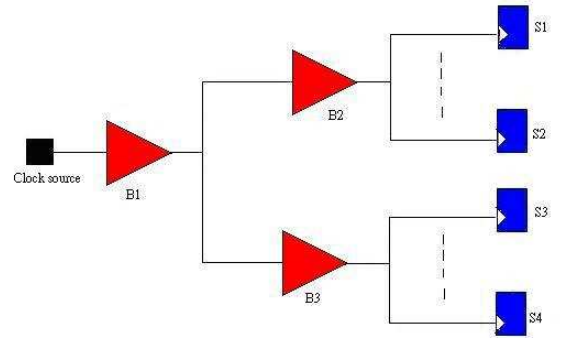


Fig. 1. A buffered clock tree with synchronizing elements  $\{S1, S2, S3, S4\}$  and buffers  $\{B1, B2, B3\}$ .

number of buffers required in the clock tree given a maximum transition time constraint. Moreover, [9] emphasized that the transition time becomes the key factor of low power clock design and the trade-off between the power and transition time when optimizing the clock tree is discussed. However, none of them can effectively use buffer library to help reduce power dissipation while inserting buffers in clock network.

In this paper, we reduce the power consumption of buffered clock tree by finding the best transition time for any given buffer library. We actually obtain up to 13% power saving based on the benchmarks from industry [1]. The remainder of the paper is organized as follows. In Section II, we briefly describe the problem of low power clock tree synthesis and estimation models. In Section III, we present our approaches on how to reduce power consumption by some observations, including buffer library analysis. We show the experimental results in Section IV and conclude the paper in Section V.

## II. LOW POWER CLOCK TREE SYNTHESIS

Clocking power becomes an addressable factor as the working frequency increases. In typical cell-based synchronous designs, clocking network contributes 20-30% of the total power. Clock tree synthesis is usually performed after cell placement to get more accurate physical information. The clock latency of a flip-flop's clock pin is the path delay starting from the root of clock tree, through the distribution cells, and ending at the clock pin (sink). In other words, the arrival time

of a flip-flop's clock pin is its clock latency. We denote all sinks of the circuit as  $S=\{s_1, s_2, \dots, s_n\}$ . A clock tree  $T_S$  is an embedding of the connection topology in chip. The skew constraint of  $T_S$  is then defined as the maximum difference of clock latency, which is:

$$Skew(T_S) = \max\{t(s_0, s_i) - t(s_0, s_j)\} \forall s_i, s_j \in S \quad (1)$$

where  $t(s_i, s_j)$  is propagation delay from  $s_i$  to  $s_j$  and will be explained in the following subsection. We then present our power estimation model and formulate our problem.

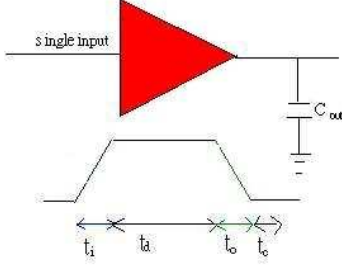


Fig. 2. Input transition time and output loading sensitive delay model in delay estimation.

### A. Delay Estimation

Fig. 2 illustrates a more abstract delay model. The time  $t$  needed for a signal to propagate from the input of a gate to the input of the next gate is  $t=t_d+t_o+t_c$ , where  $t_d$  is the propagation time inside the gate,  $t_o$  is the output transition time, and  $t_c$  is the connection time spent in the wire. The delay depends on the output load and on the input transition time.

The most accurate delay estimation is a simulation from differential equations, e.g., with SPICE. However it is too much CPU expensive. A table lookup based nonlinear interpolation delay model is within 3% of SPICE [3]. We therefore use such an accurate table lookup based nonlinear delay model. For two-dimensional tables, the delay calculator uses a bilinear interpolation routine to determine these intermediate function values. For example, in Fig. 3, we are to compute a cell's delay value  $D_{target}$  for a specific effective load  $C_{target}$  and input transition time  $T_r$ , delay calculator uses the delay values  $D_1, D_2, D_3$  and  $D_4$  of the four points in the table whose loading and input transition time come closest to the specified load and input slew values. To compute  $D_{target}$ , delay calculator determines  $Temp_A$  and  $Temp_B$  which are intermediate values computed by linear interpolation. The equations for delay calculator to calculate  $D_{target}$  are as follows.

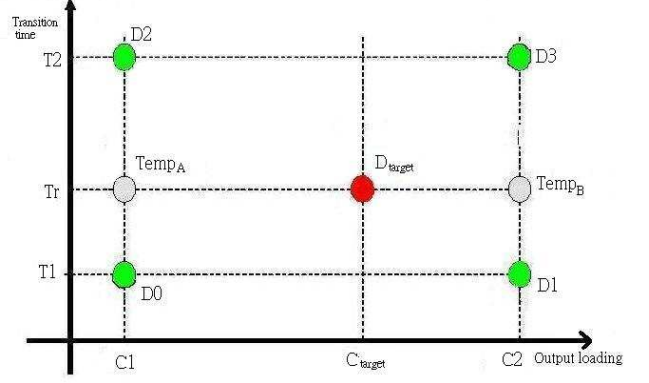


Fig. 3. Interpolation of cell delay calculation.

$$Temp_A = D_0 + \frac{T_r - T_1}{T_2 - T_1}(D_2 - D_0) \quad (2)$$

$$Temp_B = D_1 + \frac{T_r - T_1}{T_2 - T_1}(D_3 - D_1) \quad (3)$$

$$D_{target} = Temp_A + \frac{C_{target} - C_1}{C_2 - C_1}(Temp_B - Temp_A) \quad (4)$$

### B. Power Estimation

To optimize the power for clock tree, it is important to have a legitimate estimate of power consumption. In a clock tree, wires and cells contribute to the power consumption. For wires, the power  $P_{sw}(wire) = fC_{wire}V^2$  is dissipated by charging and discharging the wire capacitance. We use the equation below to estimate the wire capacitance for calculating the delay of clock buffers. The equation to estimate the wire load  $C$  of a driver pin is

$$C_{wire} = \sum_{all\_fanout} wirelength * \phi \quad (5)$$

where  $\phi$  is the weighting parameter from [1]. We can then obtain the wirelength by summing up the Manhattan distance of any two connecting cells. The power consumption of cells consists of two components. One is switching power consumption  $P_{sw}(cells)$  which corresponds to charging and discharging of the capacitance in cells. The other is an internal power (short-circuit power) of cells. We use table lookup based nonlinear power model library in this paper to find accurate values of internal power for cells. The estimation of total power is then from the switching power (wire and cells) and internal power  $P_{int}(cells)$  (short-circuit power) of cells:

$$P_{total} = P_{sw}(cells) + P_{sw}(wires) + P_{int}(cells) \quad (6)$$

### C. Problem Formulation

**Problem 1 Low Power Buffered Clock Tree Synthesis:** Given a set of sinks (flip-flops) of the circuit as  $S=\{s_1,s_2,\dots,s_n\}$  and buffer library, construct a buffered clock tree which minimizes clock skew while reducing the power consumption on cells (flip-flops and clock buffers) and wires (wirelength).

## III. LOW POWER BUFFERED CLOCK TREE SYNTHESIS BY CUSTOMIZED BUFFER INSERTION

Due to skew constraints, buffer insertion becomes necessary in clock tree synthesis. However, this technique may lead to more power penalty. Through effective analysis, we have made some observations for buffer insertion techniques in reducing clock power.

### A. Use Smaller Buffer Type

Why the smaller type is better than the bigger one in buffer insertion? Previous works emphasize that they can obtain optimal buffer size and insertion solution but they are not fitted into clock tree construction. If we choose smaller buffer type instead of the bigger one, we can obtain shorter total wire lengths between buffers and standard cells. In Fig.4(a) and Fig.4(b) show different wirelength results. In Fig.4(a), we insert two small buffers and generate two clusters in the circuit. However in Fig. 4(b) we insert the buffer which has two times the size of small one, we observe longer wirelength. Therefore we can reduce switching power component for wires by this observation.

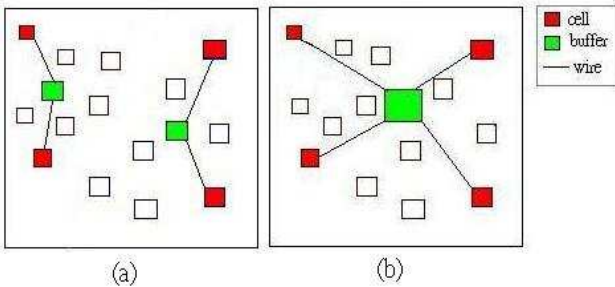


Fig. 4. Consider two cases with different size of buffers, shown in Fig.4(a) and Fig.4(b). We can reduce switching power component for wires by inserting smaller buffers(Fig.4(b)).

### B. Prefer Inverter When Inserting Clock Buffers

From the buffer library that includes clock buffers and inverters, we prefer inserting inverters since we find that the power consumption of inverter is less than that of buffer type. Previous works seem not to use inverter as repeater since they do not mention that there exists inverter type in buffer library. And there exists a critical problem in

using inverter insertion: phase problem. The function of the circuit could be incorrect after clock tree construction. In order to avoid this problem, we need to keep an even number of inverters from the root of clock tree to the leaf of the tree. We solve it by requiring that we insert inverters at each level of clock tree, then decide to insert the buffer or the inverter in the root of clock tree.

### C. Clustering of Clock Tree for Load Balancing

We have implemented a clustering algorithm that can be used at different levels of a clock tree. At bottom level, the algorithm clusters flip-flops, while at middle levels of the clock tree, clock buffers are clustered. The input of the algorithm is a set of flip-flops or clock buffers with associated locations and input capacitances, and the size of cluster loading (from next subsection). The task is to partition the input set into  $K$  clusters such that all of them have approximately the same load, then equally-sized buffers at the same level are inserted. Interconnect delays within clusters are concurrently balanced as well, thereby generating a low-skew buffered clock tree design. The clustering is illustrated in Fig.5.

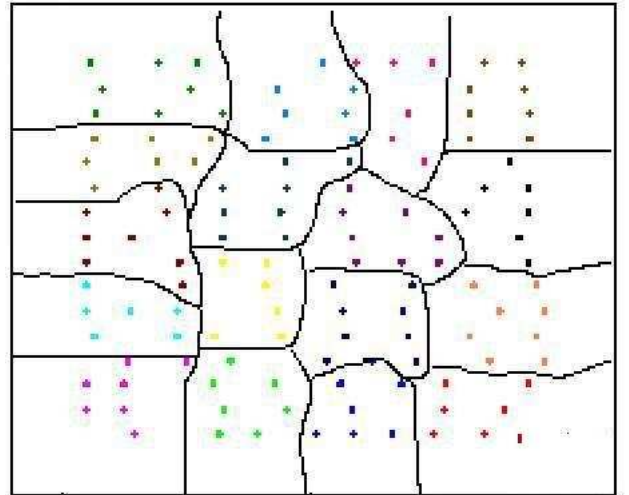


Fig. 5. Flip-flops are partitioned into clusters of approximately identical loading.

### D. Find Best Transition Time Between Buffers and Flip-Flops

Not only can we reduce total power consumption by reducing total wirelength, but we can also benefit from analyzing output loading and transition time between buffers and flip-flops. Based on our observation, we can analyze the given buffer library and use the relationship between transition time of buffers and power consumption reduction. At the first level of clock tree, buffers need to drive some flip-flops. We observe that cells will consume power differently under different transition time.

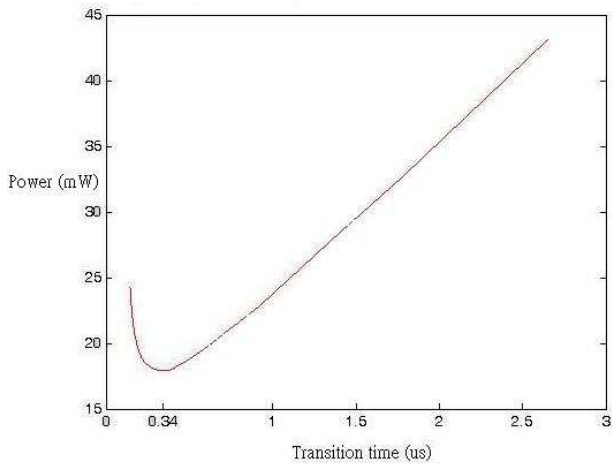


Fig. 6. Different power dissipation in a cluster can be obtained due to different transition time between buffer and flip-flops. We analyze the buffer library and get the best transition time to save power.

From given buffer library, we understand the characteristics for buffers and flip-flops such as power, delay and transition time. We then analyze corresponding power for buffers driving flip-flops. We further know that under the best transition time, less power consumption can be achieved. For example, there is a circuit with 123 flip-flops and we choose one type of inverters in buffer library for repeater insertion. The simple correspondence relation is shown in Fig.6. We cluster flip-flops according to the result that we analyze from buffer library. We can generate clusters of nearly identical loading and drive each of these clusters with identical inverters (Fig.5).

### E. Overlap-Free Clock Buffer Placement

It is not feasible when inserting inverters/buffers on preoccupied space. We use a simple approach and it can be explained by illustrated example in Fig.7. If we find that inserted clock buffer overlapping over other cells, we consider the feasible range (shown in Fig.7(a)). We choose a feasible position around the cell in this feasible range, then place the buffer (shown in Fig.7(b)).

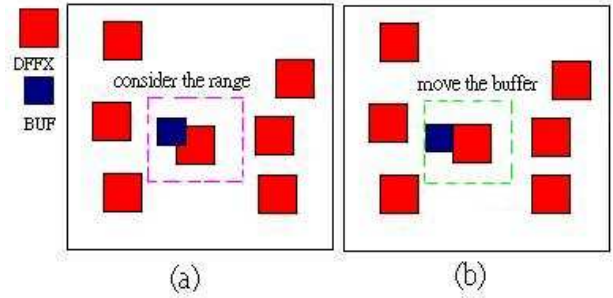


Fig. 7. Overlap-free clock buffer placement.

## IV. EXPERIMENTAL RESULTS

We have applied the above design methodology to the benchmarks in [1]. First we automatically analyze buffer library and obtain the characteristics of each buffer. The number of clusters is then based on the best loading in the circuit. We obtain each cluster of approximately equal capacitance loading and further insert buffers to drive identical loading at the same level of clock tree. Our design methodology is shown in Fig.8.

In order to show the effectiveness of our approach, we also im-

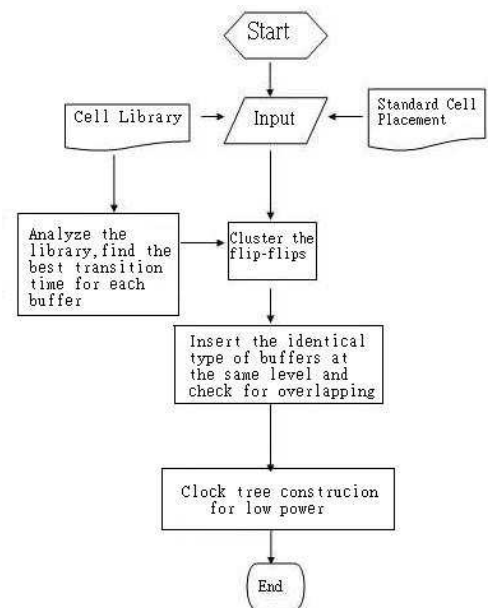


Fig. 8. Design flow for low power buffer clock tree synthesis considering best transition time.

TABLE 1  
COMPARISON OF THE EXPERIMENTAL RESULTS FROM THE APPROACH IN [3] AND OURS. OUR APPROACH OBTAIN BETTER POWER SAVING RATIO AND MUCH SMALLER CLOCK SKEW.

Benchmark	# of flip-flops	Gate sizing	Our approach	reduction ratio(%)	Gate sizing	Our approach	reduction ratio(%)
		clock skew(ns)			power(mW)		
Clk_100	123	0.00849	0.00638	24.9%	4.48512	4.13609	8.4%
Clk_500	500	0.03117	0.01867	40.1%	19.29948	17.51394	10.1%
Clk_1000	1234	0.05046	0.03454	31.5%	46.80549	42.30692	10.4%
Clk_5000	5000	0.1872	0.05682	69.6%	188.39265	169.13925	11.38%
Clk_10000	10000	0.21826	0.09648	55.8%	377.51768	331.92524	13.73%
Average				44.4%			10.8%

plement another approach [3] which apply gate/buffer sizing techniques. In Table 1 we show the experimental results of power reduction ratio and those outperform the results from [3]. Buffer library analysis finds the best transition time to optimize power dissipation, this means we make use of each buffer efficiently. Moreover, this approach can obtain more power reduction when the number of flip-flops is getting larger. The results show that our approach can achieve low power clock design and it is scalable to larger circuits.

The results in Table 1 also show significant improvement in clock skew. It is because the approach in [3] may change the size of inserted buffers and that may cause the increase of clock skew. We select, however, to insert identical size of the buffers in the same level of clock tree, which effectively avoids the skew to increase.

## V. CONCLUSION

A buffered clock tree is often desirable to solve skew problem. However, as technology advances, chips running at higher frequency are harder to obtain lower power consumption in clock network. In this paper, we further verify that the transition time is one of the key factors for low power clock design. We propose to analyze buffer library to find best transition time in buffered clock tree synthesis. According to the result of buffer library analysis, we cluster flip-flops such that clusters are approximately the same loading. Each of these clusters is driven by identical buffer type. We have obtained averagely 10.8% power saving compared with a previous approach in aggressive gate sizing. Due to the use of equally-sized buffer at the same level of clock tree, we can generate a low-clock-skew buffered clock tree.

## REFERENCES

- [1] "<http://www.cs.nthu.edu.tw/~cad/Problems.html>".
- [2] Nan-Chi Chou and Chung-Kuan Cheng. "Wire Length And Delay Minimization In General Clock Net Routing.". In *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, pages 552–555, 1993.
- [3] Olivier Coudert. "Gate Sizing for Constrained Delay/Power/Area Optimization.". In *Very Large Scale Integration (VLSI) Systems, IEEE Transactions*, pages 465–472, 1997.
- [4] M. Edahiro. "Delay Minimization For Zero-skew Routing.". In *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, pages 563–566, 1993.
- [5] A.H. Farrahi et al. "Activity-driven clock design.". In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 705–714, 2001.
- [6] Chunhong Chen et al. "Activity-sensitive clock tree construction for low power.". In *Int'l Symp. Low Power Electronics and Design*, pages 279–282, 2002.
- [7] Ting-Hai Chao et al. "Zero skew clock routing with minimum wire-length.". In *Circuits and Systems: Analog and Digital Signal Processing, IEEE Transactions*, pages 799–814, 1992.
- [8] A.B. Kahng and Chung-Wen Albert Tsao. "Planar-DME: a single-layer zeroskew clock tree router.". In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 8–19, 1996.
- [9] Min Pan, Chris Chong-Nuen Chu, and J. Morris Chang. "Transition Time Bounded Low-power Clock Tree Construction.". In *Proceedings IEEE Asia and South Pacific Design Automation Conference*, 2005.
- [10] S. Pulella, N. Menezes, and L.T. Pillage. "Low power IC clock tree design.". In *Custom Integrated Circuits Conference*, pages 263–266, 1995.
- [11] G.E. Tellez and M Sarrafzadeh. "Minimal buffer insertion in clock trees with skew and slew rate constraints.". In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 333–342, 1997.
- [12] C.-W. A. Tsao and C.-K. Koh. "UST/DME: A clock tree router for general skew constraints.". In *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, pages 400–405, 2000.
- [13] R.-S. Tsay. "Exact zero skew.". In *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, pages 336–339, 1991.
- [14] M. Vittal, A. and Marek-Sadowska. "Low-power buffered clock tree design.". In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 965–975, 1997.
- [15] J. G. Xi and W. W.-M. Dai. "Useful-skew clock routing with gate sizing for low power design.". In *Journal of VLSI Signal Processing Systems*, pages 163–170, 1997.