Multi-bit Flip-flop Generation Considering Multi-corner Multi-mode Timing Constraint

Taehee Lee[#], JongWon Yi[*] and Joon-Sung Yang[#]

[#]Department of Semiconductor and Display Engineering, Sungkyunkwan University, Korea
*System LSI Business, Samsung Electronics Co. Ltd., Giheung, Korea
E-mail: {thl3510@skku.edu js.yang@skku.edu}

***Abstract*— Clock power is a significant portion of chip power in System-on-chip (SoC). Applying Multi-bit flip-flop (MBFF) is capable of providing attractive solution to reduce clock power. To our best knowledge, this is the first work in the literature that considers multi-corner and multi-mode (MCMM) timing constraint for the MBFF generation. This proposed method is applied to five industrial digital intellectual property (IP) blocks of state-of-the-art System-on-chip (SoC). Experimental results show that our proposed MBFF generation algorithm achieves 22% clock power reduction.**

***Keywords*— Clock Network Optimization, Clock Gating, Multi-bit Flip-Flop generation, Multi-corner Multi-mode**

I. INTRODUCTION

Power minimization is one of the foremost objectives in today mobile SoCs used in portable devices because their main source of power is a battery, which is able to supply a limited amount of energy. In the composition of the total power of mobile SoC, clock power is the major portion because the clock nets operate at the high switching frequency and the number of flip-flops and clock buffers have increased dramatically with the high degree of design complexity in mobile SoC.

Previous researches on a multi-bit flip-flop (MBFF) generation have been presented to reduce a clock power dissipation and total flip-flop area [1-8]. Merging several flip-flops into a single MBFF significantly reduce the number of inverters because one common clock inverter is shared for slave and master latches among the flip-flops in MBFF. Consequently, the MBFF generation can reduce the total power and the total area of the whole flip-flops in mobile SoC design.

The mobile SoC has to operate under multiple modes such as various functional modes, test modes and several power modes. For instance, some IPs in the mobile SoC should be in sleep or active mode when power or performance is the main objective, respectively. Furthermore, the IPs can also operate at different power modes and various frequencies with dynamic voltage frequency scaling (DVFS) schemes. In order to reduce the number of timing engineering change order (ECO) which is always used after detailed routing to fix timing violations, we should consider multi-corner multi-mode (MCMM) timing constraints. This paper proposes a MBFF generation technique considering MCMM.

To the best of our knowledge, this is the first approach in the literature which considers MBFF generation under MCMM timing constraints.

The important contributions of this paper are summarized as follows:

- MCMM constraints are tightly coupled with the proposed MBFF generation flow. The MCMM aware MBFF generation improves timing quality of result (QOR) and reduces the number of timing ECO iterations.
- Our experimental results show that the proposed approach achieves about 22% clock power reduction while satisfying MCMM timing constraints.

The rest of this paper is organized as follows: Section II discusses a MBFF generation. Section III explains the experimental results. The paper is finalized in Section IV.

II. MBFF GENERATION

A. Conventional Approach

For clock power minimization, several single-bit flip-flops are merged and form MBFF. Elmore delay model is used to convert timing slack information to a corresponding maximum allowable wire length [4,7]. The following equation shows the maximal allowable length, l under the timing constraint, t_{max} by using Elmore delay model.

$$l \le \frac{\sqrt{c_0^2 R^2 + r_0^2 C^2 + 2 r_0 c_0 t_{\max}} - c_0 R - r_0 C}{r_0 c_0} \qquad (1)$$

where c_0 and r_0 are a unit capacitance and a unit resistance, respectively. R and C are a driver strength and a driving load, respectively. According to (1), the maximum input and output wire lengths of flip-flops can be extracted. A *timing slack free region* of each flip-flop is then generated based on the maximum wire length. A flip-flop can be placed anywhere within the *timing slack free region* such that timing constraint for connected pins of the flip-flop is not violated as shown in Fig. 1. If there is an intersecting *timing slack free region* among single-bit flip-flops' *timing slack free region*, they can be merged and form a MBFF. The highlighted region in Fig. 1 is

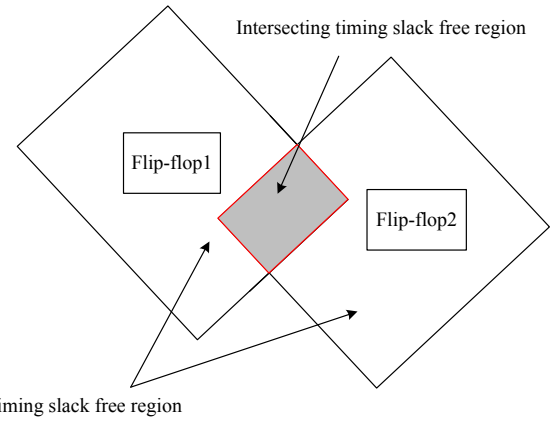
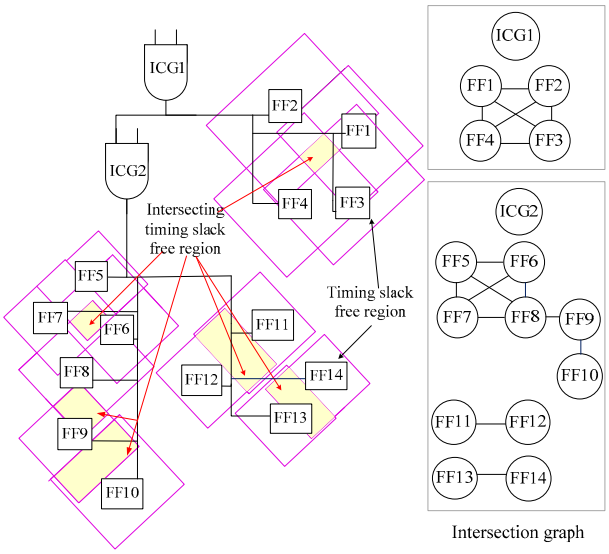
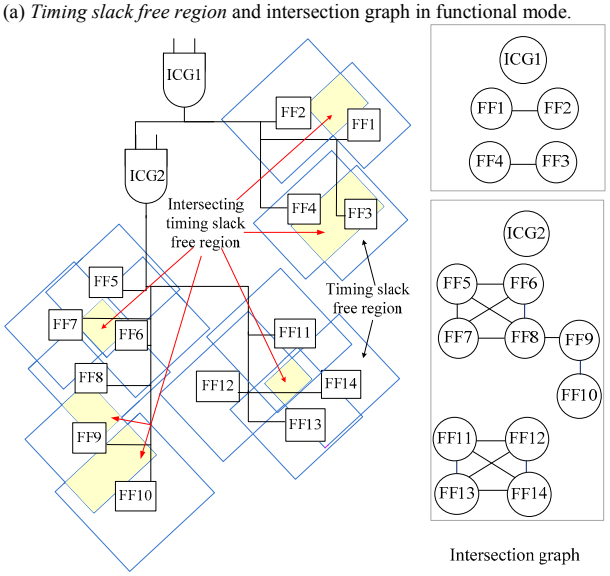


Figure 1. Timing Slack Free Region and Intersecting Timing Slack Free Region of Flip-flops.



(a) *Timing slack free region* and intersection graph in functional mode.



(b) *Timing slack free region* and intersection graph in test mode.

Figure 2. Example of *Timing Slack Free Region* and Intersection Graph for Functional Mode and Test Mode

the intersection between flip-flop1 and flip-flop2, hence, they can be replaced by a MBFF.

B. Proposed MBFF Generation under MCMM

Modern mobile SoCs dynamically change the clock frequencies depending on working scenarios and environments. If IPs operating with a wide-voltage range are only timing optimized at either high voltage or low voltage, it may not satisfy timing constraint at the other voltage mode. This could happen since a cell delay varies by process, voltage, and temperature (PVT) corners and various modes have different target clock frequency [9]. Therefore, we consider MCMM timing constraints for various modes and corners in MBFF generation. Otherwise, conventional MBFF generation approaches considering a single mode would induce timing violations at the other mode or corner. The replacement of single-bit flip-flops by MBFF is performed based on ICG. We form MBFFs only when the single-bit flip-flops belong to the same ICG. Because they operate at the same

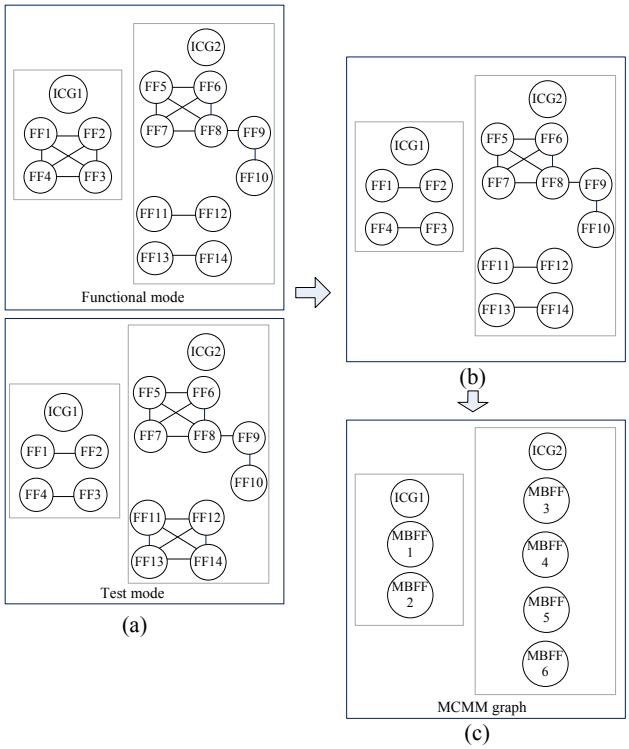


Figure 3. Merging Process of Intersection Graphs in Both Functional and Test Mode.

clock gating scheme, they can be merged and replaced by MBFF. For example, if two flip-flops are driven by different ICGs and their clock gating enable signals are different, the flip-flops cannot be merged as MBFF.

To perform the proposed MBFF generation flow, a *timing slack free region* is defined for each single-bit flip-flop and *intersecting timing slack free region* among them are found. They can be represented as a graph. Each ICG constructs an intersection graph, $G(V, E)$, whose vertices (V) represent single-bit flip-flops and edges (E) describe the *intersecting timing slack free region*s between flip-flops. In Fig. 2(a), two intersection graphs, $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, are created. For ICG1, $G_1(V_1, E_1)$ is constructed where $V_1 \in \{FF1, FF2, FF3, FF4\}$ and E_1 with { $\{FF1, FF2\}$, $\{FF1, FF3\}$, $\{FF1, FF4\}$, $\{FF2, FF3\}$, $\{FF2, FF4\}$, $\{FF3, FF4\}$ }. $G_2(V_2, E_2)$ is created for ICG2 where $V_2 \in \{FF5, FF6, FF7, FF8, FF9, FF10, FF11, FF12, FF13, FF14\}$ and E_2 with { {FF5, FF6 }, {FF5, FF7}, {FF5, FF8}, {FF6, FF7}, {FF7, FF8}, {FF6, FF8}, {FF8, FF9}, {FF9,FF10}, {FF11, FF12}, {FF13, FF14} }.

Note that the proposed method performs MBFF generation under MCMM. Because the *timing slack free region*s and intersecting region of flip-flops in an IP are different depending on various modes such as functional modes and test modes, the intersection graph is subject to each mode. For example, in Fig. 2(a), assume that there is one IP operating at 130MHz in functional mode and the fanouts of FF1, FF2, FF3 and FF4 are declared as a multi-cycle path in timing constraints. Fig. 2(b) shows the same IP in test mode targeting for 100MHz which is slower than functional mode. However, the timing constraints for FF1, FF2, FF3 and FF4 are much tighter than functional mode since they have a single-cycle path in the test mode. The *timing slack free region* of FF1, FF2, FF3 and FF4 in test mode is smaller than functional mode.

Fig. 2 shows an example with two modes – functional mode and test mode. Fig. 2(a) and (b) show a MBFF generation for functional

mode and test mode respectively. We generate intersection graphs for each mode, hence, there are $G_{fm\_1}(V_1, E_1)$, $G_{fm\_2}(V_2, E_2)$, $G_{tm\_1}(V_1, E_1)$, and $G_{tm\_2}(V_2, E_2)$. G_{fm_1} is a graph for functional mode with ICG1, G_{fm_2} is for functional mode with ICG2, G_{tm_1} is a graph for test mode with ICG1, and G_{tm_2} is with test mode under ICG2. The following shows the vertices and edges for each graph.

$G_{fm\_1}(V_1, E_1)$: $V_1 \in$ {*FF1, FF2, FF3, FF4*}
$E_1 \in$ { {*FF1, FF2*}, {*FF1, FF3*}, {*FF1, FF4*}, {*FF2, FF3*}, {*FF2, FF4*}, {*FF3, FF4*} }

$G_{fm\_2}(V_2, E_2)$: $V_2 \in$ { *FF5, FF6, FF7, FF8, FF9, FF10, FF11, FF12, FF13, FF14*}
$E_2 \in$ { {*FF5, FF6*}, {*FF5, FF7*}, {*FF5, FF8*}, {*FF6, FF7*}, {*FF7, FF8*}, {*FF6, FF8*}, {*FF8, FF9*}, {*FF9,FF10*}, {*FF11, FF12*}, {*FF13, FF14*} }

$G_{tm\_1}(V_1, E_1)$: $V_1 \in$ {*FF1, FF2, FF3, FF4*}
$E_1 \in$ { {*FF1, FF2* }, {*FF3, FF4*} }

$G_{tm\_2}(V_2, E_2)$: $V_2 \in$ { *FF5, FF6, FF7, FF8, FF9, FF10, FF11, FF12, FF13, FF14*}
$E_2 \in$ { {*FF5, FF6* }, {*FF5, FF7*}, {*FF5, FF8*}, {*FF6, FF7*}, {*FF7, FF8*}, {*FF6, FF8*}, {*FF8, FF9*}, {*FF9,FF10*}, {*FF11, FF12*}, {*FF11, FF13*}, {*FF11, FF14*}, {*FF13, FF12*}, {*FF13, FF14*}, {*FF12, FF14*} }

Once all intersection graphs are generated for ICGs and MCMM, we overlap the graphs under each ICG. For each ICG, there are multiple graphs by MCMM and they are overlapped to find a list of single-bit flip-flop candidates for MBFF. Fig. 3(a) shows the ICG1 and ICG2 graphs generated in Fig. 2 for functional and test modes. The multiple graphs are overlapped and this provides flip-flop candidates for MBFF in Fig. 3(b). This updates the graphs as follows:

$G_1(V_1, E_1)$: $V_1 \in$ {*FF1, FF2, FF3, FF4*}
$E_1 \in$ { {*FF1, FF2* }, {*FF3, FF4*} }

$G_2(V_2, E_2)$: $V_2 \in$ { *FF5, FF6, FF7, FF8, FF9, FF10, FF11, FF12, FF13, FF14*}
$E_2 \in$ { {*FF5, FF6* }, {*FF5, FF7*}, {*FF5, FF8*}, {*FF6, FF7*}, {*FF7, FF8*}, {*FF6, FF8*}, {*FF8, FF9*}, {*FF9,FF10*}, {*FF11, FF12*}, {*FF13, FF14*} }

Once the graphs are updated, we perform a MBFF generation. Based on the flip-flop clustering algorithm [4, 8], the maximal cliques are found in the updated graphs. Fig. 3 (c) illustrates an output by the maximal clique algorithm. As can be seen, ICG1 MCMM graph generates two 2-bit MBFFs by merging FF1-FF2 and FF3-FF4. For ICG2, one 4-bit MBFF and three 2-bit MBFFs are generated by merging FF5-FF6-FF7-FF8, FF9-FF10, FF11-FF12 and FF13-FF14, respectively.

III. Experimental Results

For experiments, the five industrial IPs in state-of-the-art mobile SoCs are used. Table I shows the target frequencies for five IPs. The first column shows IPs. To show MCMM, the target frequencies for two operating voltages (0.8V and 0.9V) and two modes (test mode and functional mode) are given in MHz. As can be seen, the target frequencies for MCMM are different. Hence, MBFF generation should be performed considering MCMM.

The proposed MBFF generation design flow with MCMM timing constraint is compared with a reference design flow. The proposed design flow and a reference design flow are as flows:

(1) Design Flow with MBFF generation in Single Mode : MBFF generation is applied based on INTEGRA[5]. It considers only 0.8V functional mode timing constraint.

TABLE I
Target Frequency of Five IPs in Each Mode

IP	Target Clock Frequency (MHz)			
	Test mode Operating Voltage = 0.9V : Case 1	Test mode Operating Voltage = 0.8V : Case 2	Functional mode Operating Voltage = 0.9V : Case 3	Functional mode Operating Voltage = 0.8V : Case 4
IP1	460	350	460	350
IP2	500	400	500	400
IP3	600	450	600	450
IP4	650	500	650	500
IP5	650	450	650	450

(2) Proposed Design Flow with MBFF generation in MCMM : This considers MBFF generation with MCMM timing constraints.

When comparing with design Flow with MBFF in Single Mode (1) based on INTEGRA, the proposed flow employs MCMM timing constraint to protect timing violations across multi-corners and multi-modes during multi-bit flip-flop generation in Section II.

Table II shows timing comparisons between design flow with MBFF generation in single mode (1) and proposed design flow with MBFF generation in MCMM (2). A worst negative slack (WNS) and total negative slack (TNS) are measured. First, for comparison, the design flow with MBFF (1) is performed considering only a functional mode at 0.8V. The second major column shows how many WNS and TNS are measured by single mode and single voltage. The third major column shows WNS and TNS measured when the proposed method is used for MBFF generation in MCMM. As addressed in Section II, timing critical paths can be different between test mode and functional mode. In MCMM, the critical path optimization in functional mode can somewhat bring negative influence on the timing of critical path in test mode or vice versa. The timing result of functional mode at 0.8V in MCMM (Column 16 and 17) is slightly worse than the timing result in functional mode at 0.8V in single mode (Column 8 and 9). However, other modes except functional mode at 0.8V achieves significantly better timing quality of result (QOR) under MCMM timing constraints compared to the timing result with a single mode consideration. Unless MCMM timing constraint is considered, we need more ECOs after routing to fix timing related problems. In the worst case scenario, we may not be able to fix some timing violations using ECO and this ends up with an IP performance degradation.

Table III shows power and area comparisons for two design flows with five IPs. Before MBFF generation is applied, the total standard cell area is shown in the second column. In the first columns of the second, and third major column, the area for other design flows is given as a ratio divided by the area of original design without MBFF. The design flow with MBFF in single mode (1) has on-average 10% area reduction, because the number of flip-flops and clock inverters are reduced by MBFF generation. Comparing to single mode, the area increases by 1 ~ 2% in MCMM. As a result, the proposed method achieves around 9% of area reduction for all IPs. The clock power number is not provided since it is not allowed to be open in a public domain. Instead, the relative clock powers are given as a ratio divided by the power of original design without MBFF. Even though proposed design flow with MBFF in MCMM (2) consumes little bit more clock power than the design flow with MBFF in single mode (1), the proposed design flow has much smaller amount of timing violations than the design flow (1).

TABLE II
Comparison of Worst Negative Slack (WNS) and Total Negative Slack (TNS) between Design Flow with MBFF Generation in Single Mode (1) and Proposed Design Flow with MBFF Generation in MCMM (2)

IP	Design Flow with MBFF Generation in Singe Mode (1)								Design Flow with MBFF Generation in MCMM (2)							
	Test mode Operating Volt =0.9V		Test mode Operating Volt =0.8V		Functional mode Operating Volt =0.9V		Functional mode Operating Volt =0.8V		Test mode Operating Volt =0.9V		Test mode Operating Volt =0.8V		Functional mode Operating Volt =0.9V		Functional mode Operating Volt =0.8V	
	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)
IP1	-0.30	-47.40	-0.24	-32.40	-0.21	-31.29	-0.12	-1.92	-0.06	-1.16	-0.10	-1.38	-0.08	-1.53	-0.12	-1.98
IP2	-0.26	-35.88	-0.22	-32.34	-0.06	-1.85	-0.09	-1.89	-0.02	-0.41	-0.06	-1.05	-0.02	-0.68	-0.10	-1.93
IP3	-0.33	-54.12	-0.32	-50.56	-0.23	-37.86	-0.12	-4.20	-0.07	-2.55	-0.09	-2.48	-0.08	-3.45	-0.12	-4.10
IP4	-0.27	-41.85	-0.26	-38.48	-0.19	-25.34	-0.10	-4.50	-0.04	-2.29	-0.07	-2.77	-0.06	-2.21	-0.11	-4.41
IP5	-0.42	-106.18	-0.35	-79.45	-0.36	-85.11	-0.20	-10.00	-0.10	-5.07	-0.16	-6.88	-0.09	-5.01	-0.22	-10.36

TABLE III
Comparisons of Power and Area Ratio for Two Design Flows

IP	Without MBFF Generation	Design Flow with MBFF in Single mode (1)		Design Flow with MBFF in MCMM (2)	
	Total Standard Cells Area (um2)	Area Ratio	Clock Power Ratio	Area Ratio	Clock Power Ratio
IP1	362167	0.89	0.74	0.904	0.764
IP2	548388	0.9	0.76	0.911	0.778
IP3	802333	0.88	0.73	0.899	0.756
IP4	1203010	0.92	0.77	0.94	0.804
IP5	1400340	0.9	0.76	0.924	0.8

The proposed method achieves almost 22% power reduction from all IPs. The proposed flow can reduce the number of timing ECO iterations after placement and routing and meet the overall tape out schedule successfully, because our proposed flow provides better timing result than the previous work. Consequently, the proposed design flow is very effective in reducing clock power and achieving MCMM timing closure.

IV. CONCLUSION

In this paper, MBFF generation approach is proposed. It reduces a clock power while satisfying MCMM timing constraints. Experimental results show that the proposed approach can reduce a clock tree power by around 22% and the experiments are performed with state-of-the-art IPs in SoCs. This paper shows why MCMM should be considered in MBFF generation in modern multi-voltage and multi-power mode SoC designs. Our approach is very efficient in clock power reduction without degrading circuit performance under multi-corner and multi-mode.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01058856).

References

[1] C.-C. Tsai, Y. Shi, G. Luo, and I. H.-R. Jiang, "FF-Bond:Multi-bit flip-flop bonding at placement," in *Proceedings of ACM International Symposium on Physical Design*, 2013, pp. 147–153.

[2] C.-C. Hsu, Y-C. Chen, and M. P.-H. Lin, "In-placement clock-tree aware multi-bit flip-flop generation for power optimization," in *Proceedings of the IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, 2013, pp. 592-598.

[3]S-C. Lo, C-C Hsu, and M. P-H Lin, "Power optimization for clock network with clock gate cloning and flip-flop merging" in *Proceedings of International Symposium on Physical Design*, 2014, pp.77-84.

[4]S. H. Wang, Y. Y. Liang, T. Y. Kuo, and W. K. Mak, "Power-driven flip-flop merging and relocation," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 31, no. 2, 2012, pp. 180–191.

[5]I. H.-R. Jiang, C.-L. Chang, and Y.-M. Yang, "INTEGRA: Fast multibit flip-flop clustering for clock power saving," *IEEE Trans. Computer-Aided Design*, vol. 31, no. 2, 2012, pp. 192–204.

[6] S.-Y. Liu, C.-J. Lee, and H.-M. Chen, "Agglomerative-based flip-flop merging with signal wirelength optimization," in *Proceedings of IEEE/ACM Design, Automation and Test in Europe Conference*, 2012, pp. 1391–1396.

[7] Z.-W. Chen and J.-T. Yan, "Routability-driven flip-flop merging process for clock power reduction," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 2010, pp. 203–208.

[8]M. P. H. Lin, C. C. Hsu, and Y. T. Chang, "Post-placement power optimization with multi-bit flip-flops," *IEEE Trans. Comput Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, Dec. 2011, pp. 1870–1882.

[9] Y. P. J, Echeverri. M. Meijer, and J. P. D Gyvez, "Logic Synthesis of Low-power ICs with Ultra-wide Voltage and Frequency Scaling," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2014, pp.1-2.

[10] W. Shen, Y. Cai, X. Hong, and J. Hu, "Activity-aware registers placement for low power gated clock tree construction," in *Proc. IEEE Comput.Soc. Ann. Symp. VLSI*, Mar. 2007, pp. 383–388.

[11] W. Hou, D. Liu, and P. H. Ho, "Automatic register banking for low power clock trees," in *Proc. Int. Symp. Qual. Electron. Dec.*, 2009, pp 647–652.