Steiner Tree Based Rotary Clock Routing with Bounded Skew and Capacitive Load Balancing

Jianchao Lu, Vinayak Honkote, Xin Chen and Baris Taskin Electrical and Computer Engineering Drexel University, Philadelphia, PA 19104 Email: {jl597,vh32,xc43}@drexel.edu, taskin@coe.drexel.edu

Abstract-A novel rotary clock network routing method is proposed for the low-power resonant rotary clocking technology which guarantees: 1. The balanced capacitive load driven by each of the tapping points on the rotary rings, 2. Customized bounded clock skew among all the registers on chip, 3. A sub-optimally minimized total wirelength of the clock wire routes. In the proposed method, a forest of steiner trees is first created which connects the registers so as to achieve zero skew and greedily balance the total capacitance of each tree. Then, a balanced assignment of the steiner trees to the tapping points is performed to guarantee a balanced capacitive load on the rotary network. The proposed routing method is tested with the ISPD clock network contest and IBM r1-r5 benchmarks. The experimental results show that the capacitive load imbalance is very limited. The total wirelength is reduced by 64.2% compared to the best previous work known in literature through the combination of steiner tree routing and the assignment of trees to the tapping points. The average clock skew simulated using HSPICE is only 8.8ps when the bounded skew target is set to 10.0ps.

I. INTRODUCTION

Resonant rotary clocking technology is an attractive alternative to the traditional clocking due to the high frequency and low power dissipation characteristics [1-3]. However, this technology requires novel electronic design automation (EDA) routines within the physical design flow. It is desirable to have a method which efficiently encapsulates the implementation for the recently identified 1) operational characteristics such as zero clock skew synchronization [4] and 2) implementation requirements such as capacitive load balancing [1,5].

Earlier design automation methods [6,7] focus on eliminating or accounting for the non-zero clock skew as it is commonly conceived that rotary clocking only provides a non-zero clock skew synchronization. It is shown in [4] that zero skew rotary clock can be efficiently implemented with resonant clocking. However in [4], the capacitive load balancing requirement is not considered in achieving the zero skew synchronization. The rotary clock signal can be severely distorted due to the uneven capacitance distribution [1,5]. Capacitive load balancing requires a more comprehensive analysis as the tapping wires and the anti-parallel inverters on the rotary ring contribute to the capacitive load, as well. In [8], the zero skew and capacitive load balancing are considered but the method leads to undesirable amount of tapping wires.

Another major need in rotary clock routing automation is the minimization of the *tapping* wirelength used to connect

978-3-9810801-7-9/DATE11/© 2011 EDAA





(a) ROA network with registers routed individually.

(b) ROA network with registers routed in a steiner tree manner.

Fig. 1. ROA network illustration.

the synchronous components to the rotary rings. In [7], the tapping wirelength is minimized by incrementally placing the synchronous components closer to the tapping points. In [4, 8], the integer linear programming (ILP) framework is used to compute the optimal tapping wirelength given the placement of the synchronous components. In these previous works [4–8], the synchronous components are connected to the rings using individual tapping wires. In [9], it is shown that rotary clocking with the tree based sub-networks reduces the overall wirelength. However, the capacitive load balancing is not considered in building the tree sub-networks in [9].

In this paper, a novel rotary clock network routing method is proposed where the synchronous component are connected (e.g. tapped) to the rotary oscillator array (i.e. ROA) using a forest of steiner trees satisfying all objectives simultaneously. All stages of the proposed rotary clock routing technology are polynomial in time complexity. The novelties and advantages of the proposed method are:

- 1) The capacitive load driven by each ring on the rotary clock distribution grid (i.e. ROA) is balanced,
- Bounded clock skew is guaranteed under the Elmore delay model and it can be extended to other delay models,
- 3) The total wirelength connecting the tapping points to the registers is minimized by using the steiner trees.

II. PRELIMINARIES

The rotary clocking technology is traditionally implemented with a regular array (grid) topology of oscillatory rings called the *rotary oscillatory array (ROA)* [1] as shown in Figure 1. The rotary rings are generated on the cross-connected transmission lines (mobius strips) formed by regular IC interconnects. An oscillation on a rotary ring can start spontaneously upon any noise event or stimulated by a start-up circuit for controlled operation resulting in a continuous traveling wave [1]. Distributed CMOS inverters placed uniformly along the transmission lines in anti-parallel configuration adiabatically save power, amplify signals and ensure the rotational clock. Such rotary oscillator generated square waves present low jitter and controllable skew properties. The traveling oscillation of the signal around the rings causes a varying phase depending on where the clock signal is sampled. A fixed number of locations on the rings with high granularity and uniformly distributed clock signals are designated as *tapping points*.

A. Capacitive Load Balancing

The oscillation frequency of the resonant rotary clocking system is estimated as:

$$f_{osc} \approx \frac{1}{2\sqrt{L_T C_T}},\tag{1}$$

where L_T and C_T are the total inductance and total capacitance respectively, along the path of the rotary signal on the ring. The total inductance L_T depends on the rotary mobius ring geometry [1]. The total capacitance C_T is estimated by:

$$C_T = \sum C_{tra} + \sum C_{inv} + \sum C_{reg} + \sum C_{wire}, \qquad (2)$$

where C_{tra} , C_{inv} , C_{reg} , and C_{wire} are the capacitances contributed by the transmission line, inverter pairs, registers, and the register tapping wires, respectively. The operating frequency of an ROA depends on the total estimated inductance L_T and capacitance C_T in the system as estimated in (1). The capacitive load balancing of each ring is important as an imbalance can lead to a distortion in the signal waveform [5]. All the rotary rings on the ROA have the same perimeter and the structural properties. Hence, the inductance of each rotary ring on the ROA is identical. C_{tra} and C_{inv} are identical for each ROA ring. C_{reg} and C_{wire} depend on the number of registers connected to each ring as well as their physical proximity to the ring as the proximity affects the tapping wirelength (thus, C_{wire}). This potential variation on the total capacitance of each ring of an ROA negatively affects the oscillation quality [5].

B. ROA Clock Routing

The ROA clock routing involves connecting the placed registers to the tapping points on the pre-placed rings through *tapping wires* to achieve goals such as zero skew, capacitive load balancing at the tapping points and wire minimization. A previous work in [5] presents an ROA routing scheme by using an integer linear programming (ILP) formulation. Two major drawbacks in [5] are that the ILP formulation is proposed for non-zero clock skew synchronization and the registers are tapped individually to the tapping points. The previous work in [9] combines the tree network and rotary ring together to reduce the tapping wirelength. However, zero

clock skew and capacitive load balancing are not guaranteed. The previous work in [8] considers zero-skew synchronization and capacitive load balancing at the same time but the method uses excessive amount of wires such that the clock signal may be severely distorted. A post-mesh routing method is proposed in [10] in order to connect the registers to the clock mesh grids using a tree structure which is similar to the problem presented in this paper. However, this method in [10] is inapplicable to the ROA clock routing because the size of the rings in an ROA is determined by the operating frequency. Thus, the ring size may be significantly larger than a clock mesh grid, which would lead to a large clock skew if the tree structure is not built in a zero skew manner.

III. METHODOLOGY

The proposed rotary clock network routing simultaneously targets bounded skew, capacitive load balancing and wirelength minimization. The proposed method is a polynomialtime algorithm performed at the post-placement stage in the physical design flow. At this stage, an ROA is already created as described in Section II-B and the registers placement locations are known. The proposed method first clusters the registers through steiner routing while keeping zero skew within each cluster and watching the capacitance balancing of all the clusters. Then, the tapping points are connected to the sub-tree roots to achieve the goal of minimizing wirelength and maintaining the capacitive load balancing, under a bounded skew constraint.

A. Register Clustering

The unbuffered zero skew steiner tree based network generation is a well-studied problem [11–14]. The deferred merge embedding (DME) solution [14] is adapted in this work with a modified merging cost function to achieve the targeted design goals. The DME algorithm is also modified to generate a forest of unbuffered steiner trees (with multiple sub-tree roots) as opposed to a single unbuffered steiner tree.

Unlike [9], which uses min-cut partition to identify the registers that are connected to the same tree (without capacitive load balancing), a novel clustering method is proposed which achieves capacitance balancing and zero skew at the same time. The input to the proposed clustering algorithm is the placement information of the circuit and the ROA information including the number of tapping points. The number of clusters is selected to be equal to the number of tapping points so that a one-to-one connection can be derived for the purpose of capacitive load balancing. Thus, the desired features of the proposed clustering algorithms are:

- 1) The estimated total capacitance of each cluster should be balanced,
- 2) The registers in each cluster should have zero skew,
- The total wirelength used to connect the registers inside each cluster should be minimized.

A method is proposed to generate a forest of steiner trees as the register clusters which achieves the three goals. In the forest, each register cluster is connected through a steiner

Algorithm 1 Register clustering (Bottom up phase).

- **Input:** Register locations $s_i(x, y)$ for each register s_i and number of available tapping points N.
- Output: The set of sub-tree roots *R*.
- 1: Initialize the sub-tree roots set $R = \{s_1, s_2, ..., s_n\};$
- 2: while |R| > N do
- 3: Find the sub-tree roots s_i and s_j in R such that the newly merged sub-tree s_v has the least total capacitance;
- 4: Create new sub-tree s_v by merging sub-trees s_i and s_j ;
- 5: $R = R \{s_i\}, R = R \{s_j\}, R = R \cup s_{\nu};$
- 6: end while

Fig. 2. The illustration of tapping wire connection.

equation:

$$\frac{r_0c_0}{2}l_{ij}^2 + r_0C_il_{ij} + d_i + p_j - D = 0.$$
(4)

If the target delay D is too big, a lot of extra wiring is required and the total wirelength increases. If the target delay D is too small, there might be no solution for the one-to-one matching of sub-tree roots to the tapping points guaranteeing zero skew. To this end, an optimal solution is proposed to calculate the optimal delay target as shown in Algorithm 2.

In order to solve the problem of zero skew synchronization and capacitive load balancing, a cost matrix C is introduced, where each element C_{ij} in matrix C represents the sum of the wire capacitance used to connect a sub-tree root i to a tapping point j and the total capacitance of the sub-tree i:

$$C_{ij} = c_0 l_{ij} + C_i. \tag{5}$$

The tapping wirelength l_{ij} is calculated using the Equation (4). There are two cases where solution l_{ij} should be discarded and the corresponding C_{ij} evaluates to an impractical case of ∞ (no connection available between sub-tree root *i* and tapping point *j*): 1. The quadratic equation does not have a positive solution; 2. The quadratic equation has a positive solution l_{ij} , but l_{ij} is smaller than the shortest possible wire necessary to connect the root of sub-tree *i* and the tapping point *j*. Both cases imply that the delay target can not be satisfied by connecting a tapping wire between the root of the sub-tree *i* and the tapping point *j*.

In Algorithm 2, a binary search is performed to solve for the optimal target delay value D^* for wirelength minimization. The perfect matching implies that under the current delay target, there exists a one-to-one routing solution for connecting each sub-tree root to a different tapping point. The parameters D_{max} and D_{min} are the maximum and minimum delays of connecting a sub-tree root *i* to a tapping point *j*, respectively. A customized integer *K* is introduced to denote the granularity of the optimality. The interval between D_{max} and D_{min} is equally partitioned into *K* small intervals, the average value of each small interval is stored in a vector D_t of size *K*. The optimal target delay D^* is the output of the algorithm.

C. Bounded Skew Cost Matrix Construction

The target delay is calculated based on the zero skew assumption. However, the routing wirelength and capacitive

tree. Similar to DME [14], the proposed method has two phases: The bottom up clustering phase and the top down embedding phase. The bottom up clustering algorithm is shown in Algorithm 1. Similar to DME [14], the merging of sub-trees guarantees the zero skew from each new subtree root to its sink registers. Unlike DME [14], where the merging cost is the distance between the sinks, the cost of merging sub-trees is selected as the total capacitance of the newly merged sub-trees. Consequently, if a newly merged subtree has the least total capacitance from root to sinks (registers) over all the other possible newly merged trees, this subtree is created. This merging scheme greedily balances the capacitance among the register clusters. Unlike DME, the bottom up clustering terminates when the number of sub-trees is equal to the number of tapping points. Next, the top down embedding [14] proceeds from each of the sub-tree roots to determine the locations of the root and the internal nodes of each sub-tree, identical to [14].

The total capacitance of each sub-tree is balanced as the merging cost selected is the total capacitance of the sub-trees formed by merging pairs during the bottom-up phase. Merging with reduced capacitance cost also helps to reduce the total wirelength as the capacitance is directly proportional to the wirelength in general.

B. Target Delay Calculation

As the root of the created sub-trees may not reside on the tapping points, extra (i.e. tapping) wires are needed to connect the tree roots to the tapping points as shown in Figure 2. Assuming the delay from the root of a sub-tree i to all its sink registers is d_i , and the phase delay of a tapping point j is p_j , the Elmore delay d_{ij} of the registers in sub-tree i when sub-tree i is connected to the tapping point j is calculated as:

$$d_{ij} = r_0 l_{ij} \left(\frac{c_0 l_{ij}}{2} + C_i\right) + d_i + p_j$$
(3)

where l_{ij} is the tapping wirelength needed to connect the root of sub-tree *i* to the tapping point *j*. C_i is the total capacitance on the sub-tree *i*. The parameter r_0 and c_0 are the unit wire resistance and unit wire capacitance, respectively. In order to guarantee zero skew among all the registers, all delays d_{ij} should be equal to a delay target *D*. The delay target *D* is used to calculate the wirelength l_{ij} by solving the quadratic Algorithm 2 Optimal target delay calculation.

Input: D_{max} , D_{min} , K. **Output:** The optimal target delay D^* . 1: Initialize $D_t[i] = D_{min} + (i + \frac{1}{2}) \frac{D_{max} - D_{min}}{K}, i \in \{0, 1, ..., K - 1\}$ 1}, $k_{max} = K - 1$, $k_{min} = 0$; 2: while $k_{min} \neq k_{max}$ do 3: $k_{median} = \frac{k_{min} + k_{max}}{2}$; Construct cost matrix C using delay target $D_t[k_{median}]$; 4: 5: if C has a perfect matching then $k_{max} = k_{median};$ 6: 7: else $k_{min} = k_{median};$ 8: end if 9: 10: end while 11: Return $D_t[k_{min}]$ as D^* ;

load balancing can be further optimized by relaxing the zero skew constraint to a bounded skew constraint. This is such for two reasons: 1. The additional wire required can be reduced by relaxing the delay requirement. 2. More tapping points and sub-tree roots connections may be available.

Assuming the skew bound is set to *B*, the delay upper bound D_u and lower bound D_l are $D^* + \frac{B}{2}$ and $D^* - \frac{B}{2}$, respectively. For illustration purpose without loss of generality, the locations of tapping points are restrained to be moving within only one edge of the rings. As shown in Figure 2, the upper left corner of the ring is set as the reference tapping point for tapping point *j* with phase p_{j0} . The tapping point can be moved within the top edge of the ring with distance *x* from the reference tapping point. A different *x* results in a different delay from the tapping point to the registers as shown in Equation (7), as well as a different phase delay p_j on the tapping point with a phase changing rate α as shown in Equation (8). The following formulation aims to find the value of *x* which minimizes l_{ij} such that the delay d_{ij} to the sink registers is within the tolerable delay range:

min
$$l_{ij}$$
 (6)

s.t.
$$d_{ij} = r_0 l_{ij} \left(\frac{c_0 l_{ij}}{2} + C_i \right) + d_i + p_j,$$
 (7)

$$p_j = p_{j0} + \alpha x, \tag{8}$$

$$l_{ij} = |x_i^r - x_{j0}^{tp} - x| + |y_i^r - y_{j0}^{tp}|,$$
(9)

$$d_{ij} \in [D_l, D_u]. \tag{10}$$

The only variable in the quadratic formulation is x. In implementation, this optimization problem with only one variable can be easily solved in constant time. Note that the value l_{ij} may be discarded based on the same criteria as explained in Section III-B. An l_{ij} will be calculated for each sub-tree i and each tapping point j to construct the bounded skew cost matrix C as explained in Section III-B.

D. Assignment Problem Formulation

The problem of balancing the capacitance on each tapping point is presented as obtaining a balanced assignment of subtree roots to tapping points where the difference between the maximum and minimum capacitive load is minimized. The problem formulation can be shown in Table I with the aforementioned difference as the minimization objective. The binary variable x_{ij} is the indicator of whether the sub-tree *i* and the tapping point *j* are connected. The constraints in Table I force each sub-tree root to connect to exactly one tapping point and a tapping point to have exactly one sub-tree connected to it. Theoretically, more than one tree can be connected to a tapping point to satisfy the design objectives. In the proposed solution, however, the number of sub-trees is selected to be identical to the number of tapping points to guarantee capacitive load balancing, thus, the perfect matching is implied. The problem formulation is known to be the balanced assignment problem (BAP) [15–17].

E. Balanced Tapping Points Assignment Algorithm

A two-step algorithm is applied to solve the problem as shown in Table I. The first step is to solve for the Linear Bottleneck Assignment Problem (LBAP) problem presented in Table II. The goal of an LBAP problem is *equivalent* to minimizing the maximum capacitive load by connecting the sub-tree roots to the tapping points. The cost matrix *C* defined in Section III-B is constructed under bounded skew constraint such that each entry C_{ij} represents the total capacitance cost (sub-tree and tapping wire) of connecting a sub-tree *i* to a tapping point *j*, when l_{ij} is feasible.

The LBAP problem shown in Table II is solved by Algorithm 3. Note that the matrix G[x] in Algorithm 3 is created based on matrix C with the same dimension such that all the edges with a weight greater than a capacitance threshold x are deleted. The algorithm utilizes a binary search to find the minimum max threshold x such that a feasible assignment exists in G[x]. A feasible assignment indicates a one-to-one perfect matching of each sub-tree root to each tapping point in the current matrix. A modified version of Hungarian algorithm [18] is applied to find an assignment in the cost matrix. The output of the algorithm is an optimal assignment (i.e. the values of x_{ij}) of the sub-tree roots to the tapping points, which indicates the connections such that the maximum capacitive load of all the tapping points is minimized.

The second step applies Algorithm 4 to obtain an optimal solution of the balanced assignment problem. In Algorithm 4, the output of Algorithm 3, which is the optimal assignment and the corresponding capacitance value, is used as the input. The algorithm iteratively decreases the difference between the min and max capacitance values while simultaneously checking

 TABLE I
 Balanced assignment problem (BAP) formulation.

min	$\max_{j}(\sum_{i} x_{ij}C_{ij}) - \min_{j}(\sum_{i} x_{ij}C_{ij})$
s.t.	$\sum_{i} x_{ij} = 1, \forall j$
	$\sum_{i=1}^{l} x_{ij} = 1, \forall i$
	$x_{ii} \in \{0, 1\}$

 TABLE II

 LINEAR BOTTLENECK ASSIGNMENT PROBLEM (LBAP) FORMULATION.

min	$\max_{j}(\sum_{\forall i} x_{ij} C_{ij})$
s.t.	$\sum_{i} x_{ij} = 1, \forall j$ $\sum_{j} x_{ij} = 1, \forall i$ $x_{ij} \in \{0, 1\}$

Algorithm 3 The bottleneck assignment algorithm [15].

Input: Cost matrix $C = [C_{ij}]_{n*n}$. **Output:** Optimum assignment permutation: $i \rightarrow \phi(i)$. 1: $C_{min} = \min_{i,j} (C_{ij}), \ C_{max} = \max_{i,j} (C_{ij})$ 2: **if** $C_{min} = C_{max}$ **then** Any assignment in C is optimal; 3: 4: else while $C' = \{C_{ij} : C_{min} < C_{ij} < C_{max}\} \neq \emptyset$ do 5: Find the median C_{median} of all the elements in C'; 6: Set_threshold(*C_{median}*, *C_{min}*, *C_{max}*); 7: end while 8: Set_threshold(C_{min} , C_{min} , C_{max}); 9: Find an assignment (permutation ϕ) in $G[C_{max}]$; 10: 11: end if **Procedure** Set_threshold(*C_{median}*, *C_{min}*, *C_{max}*) 1: Construct the cost matrix $G[C_{median}]$; 2: if $G[C_{median}]$ contains an assignment then $C_{max} = C_{median}$ 3: else 4: $C_{min} = C_{median}$ 5: 6: end if

for an assignment between the max and min values. The algorithm returns the optimal balanced assignment such that the difference between the max and min capacitance values is minimized. Therefore, the optimal capacitive load balanced solution of connecting the sub-tree roots to the tapping points is obtained that also satisfies the skew requirement without the excessive use of tapping wires.

The overall time complexity of the proposed method (including registers clustering, target delay calculation, bounded skew wirelength optimization and balanced assignment) is $O(N^4 + |S|^3)$, where S is the set of registers and N is the number of rings. The proof is omitted due to the page limit.

IV. THE RESULTS OF THE PREVIOUS WORKS

Zero (bounded) skew synchronization and capacitive load balancing for ROA have been previously studied in [4, 5, 8, 19]. In [4] and [19], the proposed methods aim to achieve zero skew and bounded skew respectively, but the capacitive load balancing is not considered. In [5], the capacitive load is balanced but the clock skew is not considered. The method in [8] is the only previous work that achieves the capacitive load balancing and zero (bounded) skew at the same time. However, the method in [8] uses excessive amount of tapping wires due to two reasons: 1. The individual connection between the

Algorithm 4 The balanced assignment algorithm [16].

Input: Cost matrix $C = [C_{ij}]_{n*n}$ and optimal permutation $i \rightarrow \phi(i)$ by solving Algorithm 3.

Output: Balanced assignment (permutation $i \rightarrow \phi^*(i)$).

- 1: $C_{min} = \min_{i} (C_{i\phi(i)}), C_{max} = \max_{i} (C_{i\phi(j)}).$ If $C_{min} = C_{max}$, return the current assignment permutation ϕ ; else go to Step 2.
- 2: Copy *C* to *C'*, delete in *C'* all elements such that $\{C_{ij} : C_{ij} \le C_{min} \text{ or } C_{ij} > C_{max}, i, j \in N\}$. If an assignment permutation ϕ exists in *C'*, if $cost(\phi) < cost(\phi^*)$, set $\phi^* = \phi$. Go to Step 1. If no assignment exists, go to Step 3.
- 3: If $C_{max} = \max_{i,j} (C_{ij})$, return the current best assignment permutation as ϕ^* . Otherwise increase C_{max} to be the next larger C_{ij} in *C*. Go to Step 2.

tapping points and the registers; 2. The attempt to balance the clock skew using tapping wires. Among the previous works targeting zero skew synchronization, the method in [4] obtains the best result in terms of total wirelength. Thus in the experimental results, the proposed method is compared against the results produced by [4] in terms of wirelength, capacitive load balancing and global clock skew.

V. EXPERIMENTAL RESULTS

The proposed steiner tree based rotary clock routing method is implemented in C++ and Matlab. The experiments are performed on the ISPD clock network contest and IBM r1-r5 benchmark circuits. An industrial 90nm technology library is used for routing and skew analysis. The rotary clock network is simulated using HSPICE to observe the global clock skew. The skew bound for the method is set to 10ps for all the benchmark circuits, which is a very limited skew range under the operating frequency of 4GHz.

After applying the proposed clustering and capacitive load balancing algorithms, the results for wirelength, capacitive load balancing, frequency variation and global clock skew are presented in Table III. By applying the proposed method, the total wirelength is reduced by an average of 64.2% compared to [4]. The wirelength savings are achieved through the steiner routing and the bounded skew constraints.

In [4], the capacitance imbalance is not considered, which is significant (87.2%) in order to keep the skew and wirelength small. Using the proposed method, two different sets of capacitance imbalance are compared. The first set is obtained through minimum cost tapping points and sub-tree roots assignment after applying the clustering algorithm, which has a capacitance imbalance of 42.4%. This result is better than [4] due to the greedy capacitance balancing effect at the clustering stage. The second set is obtained through the proposed balanced assignment algorithm (Table I), which reduces the capacitance imbalance to 19.4%. All the capacitance imbalance percentages using the proposed method are acceptable to establish and maintain a stable resonant oscillation, which is demonstrated by HSPICE simulations. It is observed in

	TABLE III				
WIRELENGTH, CAP BALANCING,	FREQUENCY	VARIATION	AND	SKEW	RESULTS

Circuits	Sinks	Rings	Total wirelength			Capacitance imbalance			Frequenc	y variation	Global skew (ps)	
			[4] (µm)	Proposed (µm)	Saving	[4]	Min-cost	Balanced	[4]	Proposed	[4]	Proposed
rl	267	5*5	166,973	94,379	43.5%	95.3%	41.2%	25.7%	0.03%	0.01%	6.8	9.7
r2	598	6*6	389,710	226,713	41.8%	96.9%	23.4%	17.0%	0.01%	0.00%	14.8	8.2
r3	862	7*7	535,055	264,274	50.6%	95.7%	29.2%	20.6%	0.02%	0.01%	13.2	8.4
r4	1903	9*9	1,197,450	611,484	48.9%	95.2%	24.1%	20.5%	0.02%	0.01%	20.6	8.5
r5	3101	10*10	1,968,735	924,669	53.0%	95.5%	23.8%	21.4%	0.01%	0.02%	21.3	12.5
ispd01	1107	5*5	913,626	355,340	61.1%	90.6%	30.2%	30.0%	10.40%	0.01%	—	6.5
ispd02	2249	5*5	2,773,506	706,885	74.5%	73.8%	33.0%	33.0%	43.60%	0.01%	—	12.1
ispd03	1200	5*5	351,704	78,907	77.6%	42.9%	46.7%	16.0%	33.60%	0.00%	—	9.1
ispd04	1845	5*5	474,988	109,859	76.9%	90.4%	45.7%	17.1%	0.20%	0.01%	26.0	9.0
ispd05	1016	5*5	245,857	60,557	75.4%	90.2%	66.1%	16.7%	0.00%	0.00%	13.8	10.4
ispd06	981	5*5	187,582	51,871	72.3%	99.0%	58.6%	16.1%	0.08%	0.00%	14.4	4.1
ispd07	1915	5*5	478,264	77,351	83.8%	99.4%	54.1%	26.7%	24.20%	0.01%	—	9.2
ispd08	1134	5*5	214,556	52,468	75.5%	99.7%	61.7%	8.7%	0.01%	0.01%	9.0	6.1
Average					64.2%	87.2%	42.4%	19.4%			15.5	8.8

the HSPICE simulations that all the ROA routings generated by the proposed method have very low frequency variation. However, for the routing method in [4], four benchmark circuits (*ispd01*, *ispd02*, *ispd03*, *ispd07*) have a large frequency variation due to capacitance imbalance.

The *HSPICE* simulations are performed to report the global clock skew of the ROA networks. The clock skew of the ROA rings generated by the proposed method is compared against that of the ROA rings generated by the methodology in [4]. The average skew on all the benchmark circuits using the proposed method is only 8.8ps, smaller than the average skew values produced by [4] (15.5ps). This is such as the total capacitive load of each ring using the proposed method is much smaller and less imbalanced. Note that for the benchmark circuits *ispd01*, *ispd02*, *ispd03* and *ispd07*, the ROA routings generated using the method in [4] have significant imbalances of the load capacitance, which causes a severe frequency variation. Thus, the skews of these four benchmark circuits can not be reported.

VI. CONCLUSION

In conclusion, a clock network routing method is proposed for resonant rotary clock that considers all operational characteristics and implementation requirements. The registers are routed using a forest of steiner trees while guaranteeing zero (bounded) skew and capacitive load balancing of each steiner tree. A balanced assignment algorithm is applied to connect the tapping points to the steiner trees to maintain the capacitive load balancing of each tapping point and the bounded clock skew of all the registers. The total wirelength is minimized and the method is 64.2% better in terms of the wirelength than the best known previous work in literature. The simulated skew is only 8.8ps, confirming the high expectations from this technology.

REFERENCES

- J. Wood, T. Edwards, and S. Lipa, "Rotary traveling-wave oscillator arrays: a new clock technology," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 11, pp. 1654–1665, Nov. 2001.
- [2] Z. Yu and X. Liu, "Power analysis of rotary clock," in *Proceedings of the IEEE Computer Society Symposium on VLSI*, May 2005, pp. 150–155.

- [3] C. Zhuo, H. Zhang, R. Samanta, J. Hu, and K. Chen, "Modeling, optimization and control of rotary traveling-wave oscillator," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2007, pp. 476–480.
- [4] V. Honkote and B. Taskin, "Zero clock skew synchronization with rotary clocking technology," in *International Symposium on Quality of Electronic Design (ISQED)*, Mar. 2009, pp. 588–593.
- [5] —, "Analysis, design and simulation of capacitive load balanced rotary oscillatory array," *Proceedings of the International Conference* on VLSI Design (VLSID), pp. 218–223, Jan. 2010.
- [6] Z. Yu and X. Liu, "Design of rotary clock based circuits," in *Proceedings* of the ACM/IEEE Design Automation Conference (DAC), June 2007, pp. 43–48.
- [7] G. Venkataraman, J. Hu, and F. Liu, "Integrated placement and skew optimization for rotary clocking," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 2, pp. 149–158, Feb. 2007.
- [8] V. Honkote and B. Taskin, "Skew-aware capacitive load balancing for low-power zero clock skew rotary oscillatory array," in *IEEE International Conference on Computer Design (ICCD)*, Oct. 2010, pp. 209–214.
- [9] B. Taskin, J. Demaio, O. Farell, M. Hazeltine, and R. Ketner, "Custom topology rotary clock router with tree subnetworks," *ACM Transactions* on *Design Automation of Electronic Systems*, vol. 14, no. 3, pp. 1–14, 2009.
- [10] R. S. Shelar, "An algorithm for routing with capacitance/distance constraints for clock distribution in microprocessors," in *Proceedings of the International Symposium on Physical Design (ISPD)*, Mar. 2009, pp. 141–148.
- [11] R.-S. Tsay, "Exact zero skew," in *IEEE International Conference on Computer-Aided Design (ICCAD)*, Nov. 1991, pp. 336–339.
- [12] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routing," in *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, June 1993, pp. 612–616.
- [13] T.-H. Chao, Y.-C. H. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Transactions* on Circuit and Systems (TCAS), vol. 39, pp. 799–814, 1992.
- [14] K. Boese and A. Kahng, "Zero-skew clock routing trees with minimum wirelength," in *Proceedings of the IEEE International ASIC Conference* and Exhibit, Sep. 1992, pp. 17–21.
- [15] R. E. Burkard, M. Dell'Amico, and S. Martello, Assignment Problems, 1st ed. SIAM, 2009.
- [16] R. E. Burkard and E. Cela, "Linear assignment problems and extensions," D.-Z. Du, P.M. Pardalos(Eds.), Handbook of Combinatorial Optimization, vol. 4, 1999.
- [17] D. W. Pentico, "Assignment problems: A golden anniversary survey," *European Journal of Operational Research*, vol. 176, pp. 774–794, January 2007.
- [18] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics (J. SIAM)*, vol. 5, no. 1, pp. 32–38, Mar. 1957.
- [19] V. Honkote and B. Taskin, "Skew analysis and bounded skew constraint methodology for rotary clocking technology," in *International Sympo*sium on Quality Electronic Design (ISQED), Mar. 2010, pp. 413–417.