

Sparse-Rotary Oscillator Array (SROA) Design for Power and Skew Reduction

Ying Teng and Baris Taskin

Department of Electrical and Computer Engineering
Drexel University, Philadelphia, Pennsylvania 19104
Email: yt74@drexel.edu, taskin@coe.drexel.edu

Abstract—This paper presents a unique rotary oscillator array (ROA) topology—the sparse-ROA (SROA). The SROA eliminates the need for redundant rings in a typical, mesh-like rotary topology optimizing the global distribution network of the resonant clocking technology. To this end, a design methodology is proposed for SROA construction based on the distribution of the synchronous components. The methodology eliminates the redundant rings of the ROA and reduces the tapping wirelength, which leads to a power saving of 32.1%. Furthermore, a skew control function is implemented into the SROA design methodology as a part of the optimization of the connections among tapping points and subtree roots. This control function leads to a clock skew reduction of 47.1% compared to a square-shaped ROA network design, which is verified through HSPICE.

I. INTRODUCTION

Resonant clocking technology is a fast-emerging, appealing alternative to the traditional clock tree network design, which provides a low power solution for clock generation and distribution at high frequencies [1–7]. Rotary traveling wave oscillator (RTWO) is a type of resonant clocking category [6]. A rotary oscillator array (ROA) is composed of identical RTWO rings structuring the global distribution network of a mesh-like, square-shaped ROA-based resonant clocking as presented in Fig. 1. The ROA-based clock network design has been well studied in recent years. The studies in [7–9] address the optimization of the local tree by generating subnetwork clock trees to the tapping points of the ROA. The study in [10] addresses the optimization of the regional clock tree in proposing custom-shaped rings. In these previous studies in literature, no attempts have been made to improve the quality of the resonant clocking through optimizing the mesh-like, square-shaped ROA, which is the global distribution network. Global network optimization of the ROA is pivotal as it dictates the quality of regional and local networks. Through global-network optimization of the ROA, the already low-power and low-skew characteristics of the rotary clocking technology can be further improved.

In this paper, a methodology for generating a novel ROA topology, which is aware of the distribution of the synchronous components, is proposed. The methodology incrementally places a full-mesh ROA closer to register-dense locations. Furthermore, the proposed ROA is constructed by eliminating the redundant rings from the full-mesh ROA structure according to

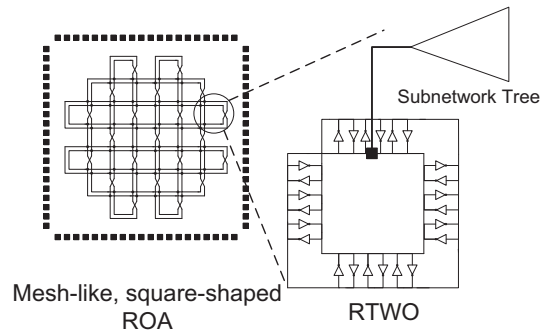


Fig. 1. ROA-based clock network.

the number and distribution of the sub-network trees. As such, this custom-shaped ROA is called the sparse-ROA (SROA). The SROA saves the total routing wirelength both in terms of the ring transmission lines (global network) and tapping wire (local network). By eliminating the redundant rings from the square-shaped ROA, the power consumption of the resonant system is reduced significantly. Furthermore, the optimization of the connections among tapping points and subtree roots both saves the tapping wirelength and decreases the clock skew among all the subnetwork trees. The ROA startup scheme in [11] is applied in order to direct the oscillation signal on the SROA to form a uniform rotation direction quickly and effectively. In conjunction with the previous literature in [7–10], the SROA design methodology presented in this paper completes the rotary-based clock network design flow by establishing the global network design step.

This paper is organized into the following sections. The ring operation is reviewed in Section II. Definitions and preliminaries for the presentation of SROA are provided in Section III. The methodology for generating the SROA is proposed in Section IV. The simulations are presented in Section V. Conclusions are presented in Section VI.

II. BACKGROUND

The regional distribution network of rotary resonant clocking, the RTWO (ring), is reviewed in Section II-A. The global network, the ROA, is reviewed in Section II-B. The ROA power consumption is discussed in Section II-C.

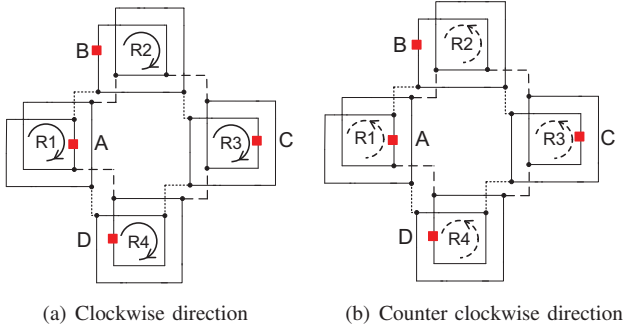


Fig. 2. ROA brick rotation direction consistency.

A. Rotary Traveling Wave Oscillator

The concept of the RTWO ring is first introduced in [6]. The traditional implementation of a single ring is a square möbius ring as shown in Fig. 1. The cross-coupled inverter pairs are used to overcome the power loss when the signal propagates along the transmission lines as well as provide a rotation lock. The phase difference from one point to another along the transmission line is proportional to the distance between these two points. The oscillation frequency of the ring is dictated by the total capacitance C_{tot} and total inductance L_{tot} of the whole ring oscillation system, which can be expressed as: $f_{osc} \approx 1/(2\sqrt{C_{tot}L_{tot}})$. In order to implement a clock with a fixed frequency, the inductance and capacitance of the ring are computed as the rings and the local subnetwork trees are being designed. The global ROA design with the rings and the subnetwork trees, that are sized for the target operating frequency, has not been previously explored.

B. Brick-based ROA

It is shown in [12] that the ROA-brick is the smallest ROA structure which guarantees that all the rings in this structure can form a uniform signal rotation direction. An ROA-brick, shown in Fig. 2, is composed of 4 rings connected in a loop structure. The feedback of the oscillation signals among the rings inside this loop enforce the signals on each ring to rotate in the same direction, either clockwise or counter clockwise. As observed in Fig. 2(a) and Fig. 2(b), regardless of the signal rotation direction, the same phase point sets (indicated by bold points labeled A, B, C, D) remain the same. In other words, zero skew clock signals are always available on these points on an ROA-brick.

C. Brick-based ROA Power Consumption

Due to its resonant property, the ROA-based clock network saves significant amount of power compared to the traditional clock generation and distribution network as shown in several studies in literature [6, 13]. The study in [13], for instance, reports that the ROA-based clock tree network can save up to 70% power consumption in a high end microprocessor compared to a conventional clock tree. Notably, the power consumption of the conventional clock network in [13] is based on estimations since it is very hard to synthesize a

clock network up to 6 – 10GHz with current processes used in academic research. In this paper, the power consumption is compared between the full-meshed ROA and SROA only.

III. DEFINITIONS AND PRELIMINARIES

The sparse-ROA (SROA) is composed of ROA-bricks. Unlike a traditional ROA, the SROA does not maintain a square structure but maintains its integrity (i.e. connectivity of RTWOs) for rotational lock between the (regional) rings of the ROA. Consequently, the SROA design methodology is synthesizing ROA bricks that are sparse but contiguous for integrity.

A. Definitions and Construction Rules of the SROA

A sample SROA topology composed of rings $\{R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11\}$ is shown in Fig. 3. Some definitions on this SROA are given below.

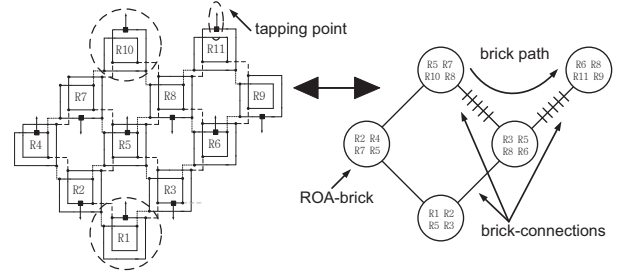


Fig. 3. A sample topology of an SROA (brick component).

- 1) **Brick-connection:** Two ROA-bricks form a brick connection if they share two rings, such as ROA-brick $\{R1, R2, R5, R3\}$ and ROA-brick $\{R2, R4, R7, R5\}$.
- 2) **Brick path:** A brick path is a sequence of ROA-bricks such that every two successive ROA-bricks form a brick-connection. For instance, Fig. 3 shows a brick path between ROA-brick $\{R5, R7, R10, R8\}$ and $\{R6, R8, R11, R9\}$ contains two brick-connections.
- 3) **Brick component:** A brick component is a group of ROA-bricks with a brick path between each pair of ROA-bricks. For instance, the topology in Fig. 3 is a brick component.
- 4) **Ring-connection:** Two ROA-bricks form a ring connection if they share only one ring (and no more) between them. For instance, ROA-brick $\{R2, R4, R7, R5\}$ and ROA-brick $\{R3, R5, R8, R6\}$ form a ring-connection.
- 5) **Ring component:** Ring component is a group of ROA-bricks, such that there is always a path formed by ring-connection(s) and/or brick-connection(s) between any two ROA-bricks. As such, a brick component is always a ring component but a ring component is not always a brick component.

The brick-connection is used for SROA synchronization. If two ROA-bricks form a brick-connection, the signal rotation direction on all the rings is easily synchronized. A ring connection, on the other hand, does not guarantee synchronization

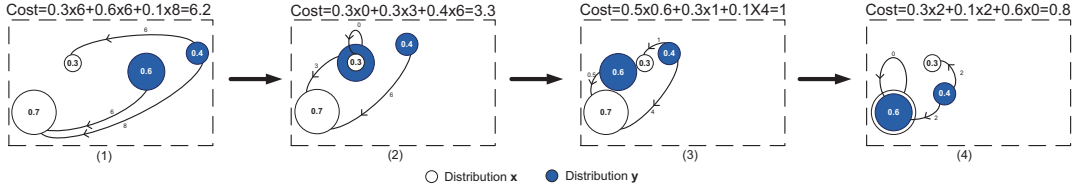


Fig. 4. The operation process of EMD_t

between ROA bricks. Based on this observation, the SROA topology is constructed with these rules:

- 1) Each ring in the SROA belongs to at least one ROA-brick.
- 2) The SROA should be a brick component.

Rule 1 indicates that every ring in the SROA should be contained in at least one ROA-brick and there should be no isolated rings in the SROA. This rule is used to maintain the integrity of the SROA. Rule 2 implies: (1) There are no disjoint brick components: If there are more than one brick components, these brick components cannot be synchronized but will operate in different frequencies. (2) Only brick-connections rather than ring-connections exist in the SROA. Rule 2 is used to direct all the rings in the SROA to form a uniform signal rotation direction. A ring-connection divides the SROA into two brick components. If these two brick components are about the same size and the signals on these components are rotating in the opposite directions, it is possible that neither brick component will be able to overwrite the signal rotation direction on the other brick component through the ring-connection. Instead of reaching synchronization, the signal undesirably dies out on the ring that connects these two brick components.

B. Earth Mover's Distance under Transformations

The earth mover's distance (EMD) is a distance measure between distributions, which is widely used in image retrieval and matching [14]. EMD under transformations (EMD_t), which is a variant of EMD that moves one of the distributions, has the advantage of computing a transformation of one distribution in order to minimize the distance (measured by EMD) between the distributions [15]. EMD_t was previously used in [16] for clocking in clock mesh automation; it is used for SROA placement in this work.

The EMD is a distance measure between discrete, finite distributions \mathbf{x} and \mathbf{y} :

$$\begin{aligned} \mathbf{x} &= \{(x_1, w_1), (x_2, w_2), \dots, (x_m, w_m)\} \\ \mathbf{y} &= \{(y_1, u_1), (y_2, u_2), \dots, (y_n, u_n)\} \end{aligned} \quad (1)$$

where, the \mathbf{x} distribution has a weight w_i at position \mathbf{x}_i in \mathbf{R}^2 , $i = 1, \dots, m$ and \mathbf{y} distribution has a weight u_j at position \mathbf{y}_j in \mathbf{R}^2 , $j = 1, \dots, n$. The EMD is defined by the linear program:

$$\begin{aligned} &\text{minimize} && \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \\ &\text{subject to} && f_{ij} \geq 0 \quad i = 1 \dots m, j = 1 \dots n \\ &&& \sum_{j=1}^n f_{ij} \leq w_i \quad i = 1 \dots m \\ &&& \sum_{i=1}^m f_{ij} \leq u_j \quad j = 1 \dots n \\ &&& \sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min(W, U) \end{aligned} \quad (2)$$

In (2), d_{ij} is distance between \mathbf{x}_i and \mathbf{y}_j . W and U are collective weights of the distributions: $W = \sum_{i=1}^m w_i$ and $U = \sum_{j=1}^n u_j$.

The earth mover's distance under transformation set \mathbf{t} (EMD_t) is defined as:

$$EMD_t(\mathbf{x}, \mathbf{y}) = \min_{t \in \mathbf{t}} EMD(\mathbf{x}, t(\mathbf{y})) \quad (3)$$

Equation (3) indicates that a transformation is performed on one distribution to move the whole distribution incrementally towards the other distribution in order to minimize the EMD . Illustration of the operation of EMD_t is shown in Fig. 4. The EMD_t is performed in two steps: 1) Assignment; 2) Transformation. As shown in Fig. 4, an assignment is performed on the two distributions \mathbf{x} and \mathbf{y} for the minimum total weighted distance (cost), which is 6.2. Then a transformation is performed on \mathbf{y} , which moves \mathbf{y} towards \mathbf{x} in order to reduce the cost. These two steps complete one iteration. The distributions in Fig. 4 require three iterations to converge to a cost of 0.8. EMD_t calculation converges very quickly. Note that, in each iteration, the assignment between two distributions may change and the transformation may also be different. The process is terminated when the assignment does not improve or the current cost is tolerable.

IV. SROA DESIGN METHODOLOGY

The design of SROA necessitates changes to local distribution networks (i.e. subnetwork trees) as the sparsity of the ROA is optimized based on the register placement. To this end, a bottom-up distribution network design procedure is proposed. First, local trees are constructed through register clustering. Second, a tapping point set is generated for optimizing the total wirelength and clock skew. Third, the SROA is generated from the full-mesh ROA based on the ring selection of the tapping point set, as well as maintaining the integrity of the SROA.

1) *Register Clustering*: In the beginning, all the registers are clustered using a modified deferred merge embedding (DME) [17–20] solution to generate a forest of unbuffered steiner tree based networks. This process is similar to the study in [7], where local tree networks are generated with BST/DME [21]. In this work, an upper bound on the total capacitance in a register cluster is established based on the target frequency of the RTWO before generating a forest of steiner trees of balanced total capacitance. Based on this capacitive limit, the subtrees are generated.

2) *Tapping Point Set Generation*: The process of tapping point set generation starts with a large enough square-shaped ROA with the number of rings larger than the number of subtree roots provided by register clustering. Each ring in the ROA provides one tapping point as a clock source. Then a modified EMD_t method is used to move the square-shaped ROA around in order to optimize the one-to-one matching between the tapping points and the subtree roots for minimum total wirelength while balancing the delay among the subnetwork trees. The tapping point set generation is shown in Algorithm 1 where the inputs are the set of the square-shaped ROA tapping points S and the set of the subtree roots D obtained from register clustering.

$$\begin{aligned} S &= \{s_1, s_2, \dots, s_m\} \\ D &= \{d_1, d_2, \dots, d_n\} \end{aligned} \quad (4)$$

For any $\mathbf{t} \in \mathbf{R}^2$, the transformation of set S is:

$$S + \mathbf{t} = \{s_1 + \mathbf{t}, s_2 + \mathbf{t}, \dots, s_m + \mathbf{t}\} \quad (5)$$

The problem is described as shown in Table I. $F = (f_{ij}) \in \mathbf{R}^{m \times n}$, with $f_{ij} = 1$ indicating a one-to-one matching from $s_i + \mathbf{t}$ to d_j . In order to keep the skew under control and to avoid undesirably large skews in individual subnetwork trees, instead of using total wirelength, the total delay is set as the optimization object of this function. As shown in Table I, the function $COST$ is to calculate the delay from $s_i + \mathbf{t}$ to a single register R_{jk} in d_j . Under the Elmore model the delay between node $s_i + \mathbf{t}$ and R_{jk} is given by:

$$\begin{aligned} t(s_i + \mathbf{t}, R_{jk}) &= \\ &|s_i + \mathbf{t} - d_j| \cdot r \left(\frac{|s_i + \mathbf{t} - d_j| \cdot c}{2} + Cap(d_j) \right) \\ &+ \sum_{e_w \in Path(d_j, R_{jk})} |e_w| \cdot r \left(\frac{|e_w| \cdot c}{2} + Cap(w) \right). \end{aligned} \quad (6)$$

Here, r and c denotes the unit length wire resistance and capacitance, respectively. $Cap(w)$ denotes the total cap seen from point w . Since the placement of the registers in each subnetwork tree are fixed before matching the tapping points to the subtree roots, the second part on right in (6) is a constant. Furthermore, since $\frac{|s_i + \mathbf{t} - d_j| \cdot c}{2} \ll Cap(d_j)$, (6) can be rewritten as:

$$t(s_i + \mathbf{t}, R_{jk}) \approx |s_i + \mathbf{t} - d_j| \cdot r \cdot Cap(d_j) + const_j. \quad (7)$$

Thus, the function $COST$, which is the sum of the total delay from set S to registers in set D under matching F and transformation \mathbf{t} , is as a weighted total wirelength from tapping points to subtree roots. $|s_i + \mathbf{t} - d_j|$ is the Manhattan distance from $s_i + \mathbf{t}$ to d_j . The total capacitance on each subtree $Cap(d_j)$ controls the skew among each subtrees. Experimental results show that $COST$ provides similar total stub wirelength compared to optimize total stub wirelength, but with bounded skew.

TABLE I
TAPPING POINT GENERATION PROBLEM.

Minimize the total matching cost under transformation.	
min	$COST(F, S, D, \mathbf{t}) =$
	$\sum_{i=1}^m \sum_{j=1}^n f_{ij} \times (s_i + \mathbf{t} - d_j \cdot r \cdot Cap(d_j) + const_j)$
s.t.	$f_{ij} = 0 \text{ or } 1, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n$
	$\sum_{j=1}^n f_{ij} \leq 1, \quad 1 \leq i \leq m$
	$\sum_{i=1}^m f_{ij} = 1, \quad 1 \leq j \leq n$
	$\mathbf{t} \in \mathbf{R}^2.$

The process is divided into two steps: (1) **best_matching**: Given the transformation \mathbf{t} , the problem becomes an integer programming problem, which is hard to solve. However, for this problem it can be mathematically proved that by relaxing the constraints $f_{ij} = 0 \text{ or } 1$ to $0 \leq f_{ij} \leq 1$, two problems have the same solution. Thus, by giving the transformation vector \mathbf{t} , linear programming can be applied to find the best matching; (2) **best_move**: Keeping the matching, the best position of the square-shaped ROA is obtained by moving it around, which leads to a minimum $COST$. In this problem, Manhattan distance is used to facilitate the calculation of the minimum total delay. The best position of the square-shaped ROA is the one with the minimum weighted total stub wirelength, the total capacitance of each subtree assists to balance the skew among all the subtrees.

The whole process is done in a number of iterations until convergence. This is because after doing the **best_move**, there may exist a better matching between $S + \mathbf{t}$ and D . The iteration ends when the improvement of $COST$ between two iterations is less than a tolerance Δ . The tapping point set contributing to the best matching is recorded as S_{opt} , the best matching result is F_{opt} and the transformation is \mathbf{t}_{opt} .

3) *SROA Generation*: The tapping point set S_{opt} generated from tapping point set generation may not be qualified to form an SROA due to a potential lack of integrity. Thus, the object of SROA Generation is to substitute a portion of the rings in the tapping point set S_{opt} with the other rings in S in order to satisfy the SROA construction rules with a minimum total distance increase. This process is outlined in Algorithm 2. The inputs are best matching F_{opt} and the best tapping point set S_{opt} from Algorithm 1. First, the largest ring component $RC_{largest}$ is investigated among S_{opt} located rings. All the ring connections in the $RC_{largest}$ need to be fixed in order to turn the $RC_{largest}$ into a brick component. For instance, in Fig. 3, ROA-bricks $\{R_2, R_4, R_7, R_5\}$ and $\{R_3, R_5, R_8, R_6\}$ only have a ring-connection between them if R_1 and R_{10}

Algorithm 1 Tapping Point Set Generation

Input: Tapping point set S and subtree root set D
Output: Optimized tapping point set S_{opt} , optimal matching record F_{opt} , the best transformation vector \mathbf{t}_{opt}

- 1: Initialize $cost = \infty$, $cost_{new} = 0$;
- 2: **while** $|cost - cost_{new}| > \Delta$ **do**
- 3: $[F, cost_{new}] = best_matching(S, D)$;
- 4: $\mathbf{t} = best_move(S, D, F)$;
- 5: $cost = cost_{new}$
- 6: $cost_{new} = COST(F, S, D, \mathbf{t})$
- 7: $S = S + \mathbf{t}$;
- 8: **end while**
- 9: $S_{opt} = S$, $F_{opt} = F$, $\mathbf{t}_{opt} = \mathbf{t}$;

do not exist. However, by adding R1 or R10, a brick path is formed. After getting $RC_{largest}$, the next step is to find all the k possible ring-connection fixing schemes. In each iteration, the $RC_{largest}$ is fixed and turns into a brick component $BC(i)$. The isolated ring set $R_{iso}(i)$ is generated:

$$R_{iso}(i) = S_{opt} \setminus BC(i). \quad (8)$$

If the ring number of $BC(i)$ is smaller than that of S_{opt} (one ring corresponding to one tapping point), the isolated rings are added back to $BC(i)$ from the one with minimum cost. In order to maintain the integrity of $BC(i)$, the isolated ring is added back by a shortest brick path. The cost is:

$$cost(i) = BC_{new}(i) \setminus BC(i) \setminus R_{iso}(i) \quad (9)$$

where $BC_{new}(i)$ is the brick component after adding back an isolated ring. Since more than one ring may be added back to $BC(i)$ at once, the cost of adding back an isolated ring is evaluated based on both the increase of non- S_{opt} located rings and the reduction of isolated rings. After adding back one isolated ring, the $BC(i)$, $R_{iso}(i)$ are updated. The process terminates when the ring number of $BC(i)$ reaches the tapping point number of S_{opt} . Then, $BC(i)$ together with D are sent to the tapping point set generation for final optimization, in case the matching and/or the location of tapping points needs to be updated. $S_{cand}(i)$ and $F_{cand}(i)$ are the best tapping point set and best matching under each ring-connection fixing scheme, respectively. The cost $cost_{cand}(i)$ under $S_{cand}(i)$ and $F_{cand}(i)$ is calculated. The index j of minimum cost among $cost_{cand}(i)$ is marked. The tapping point set cost $S_{cand}(j)$ is selected to be the tapping point set S_{sroa} of SROA. The matching $F_{cand}(j)$ of $S_{cand}(j)$ is selected to be the matching F_{sroa} of SROA.

The topology S_{sroa} satisfies the SROA construction rules that are introduced in Section III-A. First, every ring in the BC is contained in at least one ROA-brick. The isolated rings are also added back to BC by brick path. Second, under each fixing scheme, the isolated rings are always added back to the same BC . Thus, this operation guarantees that the SROA is composed by only one brick component. Furthermore, the SROA generation process also guarantees that there is no ring-connection in SROA since a BC is always maintained.

Algorithm 2 SROA Generation

Input: The best matching F_{opt} , tapping point set S_{opt} and subtree root set D
Output: The tapping points of SROA S_{sroa} , the SROA matching F_{sroa}

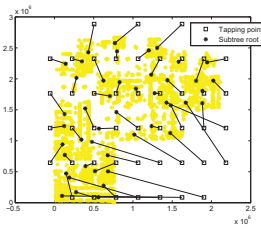
- 1: Find the largest ring component: $RC_{largest}$.
- 2: Find all the possible k ring component fixing schemes
- 3: **for** each ring component fixing scheme i **do**
- 4: Fix $RC_{largest} \Rightarrow BC(i)$
- 5: Generate $R_{iso}(i)$ based on $BC(i)$;
- 6: **while** $Sizeof(BC(i)) \leq Sizeof(S_{opt})$ **do**
- 7: Add back the minimum cost isolated tapping point;
- 8: Update $BC(i)$;
- 9: Update $R_{iso}(i)$;
- 10: **end while**
- 11: $[S_{cand}(i), F_{cand}(i), \mathbf{t}(i)] =$
 Tapping_Point_Set_Generation($BC(i), D$);
- 12: $cost_{cand}(i) = COST(F_{cand}(i), S_{cand}(i), D, \mathbf{t}(i))$;
- 13: **end for**
- 14: $cost_{cand}(j) = \min(cost_{cand})$;
- 15: $S_{sroa} = S_{cand}(j)$, $F_{sroa} = F_{cand}(j)$;

V. SIMULATIONS

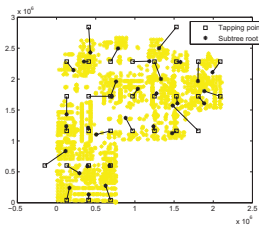
The experiments are performed on ispd10 circuit #03 – #08. The SROA structure generated for circuit #04 implemented in 90nm technology is shown in Fig. 5 in detail as an example. The operating frequency is arbitrarily picked between 6 – 7GHz. Fig. 5(a) and Fig. 5(b) show the matching of the tapping points on the regional rings of two alternative global networks: Square-shaped ROA and SROA. It is observed that the optimization on the tapping point locations of the SROA leads to a reduction in both the total stub wirelength and clock skew. For ispd10 benchmark circuit #04, 32 steiner trees are created, one of which is shown in Fig. 5(c).

Using the methodology introduced in Section IV, not only the total tapping wire is reduced but also the length of each tapping point to subtree root connection is optimized based on the subtree total capacitance. The optimization of the tapping point to subtree root connections potentially (but not with certainty as the minimum insertion delay path can be shortened, as well the maximum insertion delay path) minimizes the clock skew seen at individual sinks. After computing the structure of the clock tree network, the physical connections are transferred into a netlist and simulated using HSPICE. The power up scheme in [11] is used to facilitate the SROA synchronization process. The simulation result is shown in Fig. 6. The start-up jitter is due to the self-oscillating and phase-locking property of rotary clocking and is observed for all methods of directionality adjustment.

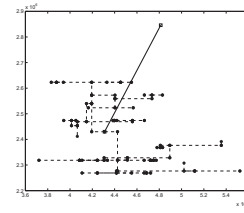
The complete test results are shown in Table II. The SROA leads to an average mesh wirelength saving of 35.3% and an average tapping wirelength saving of 26.4%. The maximum skew is consistently reduced in the SROA topology by an average of 47.1%. The average global skew of SROA-based



(a) Square-shaped ROA local subtree root connections to regional rings



(b) SROA local subtree root connections to regional rings



(c) Local subnetwork tree for one ring

Fig. 5. SROA-based network for ispd10 benchmark circuit #04.

TABLE II

SROA TESTING RESULTS COMPARED TO A TRADITIONAL ROA IMPLEMENTATION THAT CONSUMES $\approx 70\%$ LOWER POWER THAN A CLOCK TREE [13]

Circuit	ROA ring No.	SROA Ring No.	Mesh Wire Red.	Tapping Wire Red.	Power Red.	ROA skew	SROA skew	Skew Red.
ispd10 #03	29	16	44.8%	63.6%	40.7%	37.9ps	14.7ps	61.2%
ispd10 #04	55	32	41.8%	11.9%	29.7%	25.3ps	15.4ps	39.1%
ispd10 #05	55	30	45.4%	35.2%	46.0%	21.6ps	12.3ps	43.1%
ispd10 #06	21	16	19.1%	16.4%	22.8%	19.2ps	11.5ps	40.1%
ispd10 #07	37	27	27.0%	14.0%	25.8%	25.6ps	13.1ps	48.8%
ispd10 #08	28	20	28.6%	17.4%	27.5%	22.0ps	12.5ps	43.2%
Avg	-	-	35.2%	26.4%	32.1%	-	-	47.1%

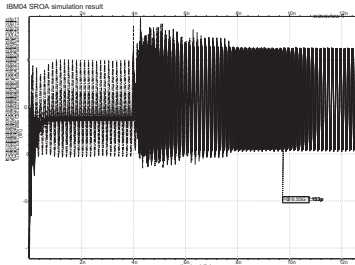


Fig. 6. HSPICE Simulation of SROA designed for ispd10 benchmark #04.

network is within $13.3ps$, which is very competitive with contemporary clocking techniques. The average power savings of 32.1% , on the already low-power ROA (Section II-C), are substantial. The combination of SROA reduction with EMD_t and regional ring reduction for SROA drives the skew reduction. The power reduction is primarily due to the reduction in the mesh ring transmission line and tapping wirelength reduction.

VI. CONCLUSION

The SROA topology is proposed for additional power savings on the low-power rotary clocking technology. The shape of the SROA is constructed based on the distribution of the synchronous components. HSPICE simulation results show that the SROA-based clock network provides an average power savings of 32.1% and, as a by product, skew reduction of 47.1% compared to the square-shaped ROA.

REFERENCES

- [1] V. Chi, "Salphasic distribution of clock signals for synchronous systems," *IEEE Trans. Comput.*, vol. 43, no. 5, pp. 597–602, May 1994.
- [2] S. Chan *et al.*, "A 4.6ghz resonant global clock distribution network," in *Proc. IEEE ISSCC'04*, Feb. 2004, pp. 342–343.
- [3] W. Andress and D. Ham, "Standing wave oscillators utilizing wave-adaptive tapered transmission lines," *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 638–651, Mar. 2005.
- [4] V. Cordero and S. Khatri, "Clock distribution scheme using coplanar transmission lines," in *Proc. DATE'08*, Mar. 2008, pp. 985–990.
- [5] A. Drake, K. Nowka, T. Nguyen, J. Burns, and R. Brown, "Resonant clocking using distributed parasitic capacitance," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1520–1528, Sept. 2004.
- [6] J. Wood, T. Edwards, and S. Lipa, "Rotary traveling-wave oscillator arrays: A new clock technology," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1654–1665, Nov. 2001.
- [7] J. Lu, V. Honkote, X. Chen, and B. Taskin, "Steiner tree based rotary clock routing with bounded skew and capacitive load balancing," in *Proc. DATE'11*, Mar. 2011, pp. 455–460.
- [8] G. Venkataraman *et al.*, "Integrated placement and skew optimization for rotary clocking," in *Proc. DATE'06*, Mar. 2006, pp. 1–6.
- [9] Z. Yu and X. Liu, "Design of rotary clock based circuits," in *Proc. ACM/IEEE DAC'07*, June 2007, pp. 43–48.
- [10] V. Honkote and B. Taskin, "CROA: Design and analysis of the custom rotary oscillatory array," *IEEE Trans. VLSI Syst.*, vol. 19, no. 10, pp. 1–11, Oct. 2010.
- [11] Y. Teng and B. Taskin, "Synchronization scheme for brick-based rotary oscillator arrays," in *Proc. ACM GLSVLSI'12*, May 2012, pp. 117–122.
- [12] Y. Teng, J. Lu, and B. Taskin, "RoA-brick topology for rotary resonant clocks," in *Proc. IEEE ICCD'11*, Oct. 2011, pp. 273–278.
- [13] Z. Yu and X. Liu, "Power analysis of rotary clock," in *Proc. IEEE ISVLSI'99*, May 2005, pp. 150–155.
- [14] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 1–20, 2000.
- [15] S. Cohen and L. Guibas, "The earth mover's distance under transformation sets," in *Proc. IEEE ICCV'99*, Sept. 1999, pp. 1076–1083.
- [16] Y. Teng and B. Taskin, "Clock mesh synthesis method using the earth mover's distance under transformations," in *Proc. IEEE ICCD'12*, Oct. 2012, pp. 121–126.
- [17] R. Tsay, "Exact zero skew," in *Proc. ACM/IEEE ICCAD'91*, Nov. 1991, pp. 336–339.
- [18] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *Proc. ACM/IEEE DAC'93*, June 1993, pp. 612–616.
- [19] T. Chao, Y. Hsu, J. Ho, and A. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Trans. Circuits Syst. II*, vol. 39, no. 11, pp. 799–814, Nov. 1992.
- [20] K. Boese and A. Kahng, "Zero-skew clock routing trees with minimum wirelength," in *Proc. IEEE International ASIC Conference and Exhibit'92*, Sept. 1992, pp. 17–21.
- [21] J. Cong, A. Kahng, C. Koh, and C. Tsao, "Bounded-skew clock and steiner routing," *ACM Transactions on Design Automation of Electronic Systems*, vol. 3, no. 3, pp. 341–388, July 1998.