College of Engineering



Drexel E-Repository and Archive (iDEA) <u>http://idea.library.drexel.edu/</u>

Drexel University Libraries www.library.drexel.edu

The following item is made available as a courtesy to scholars by the author(s) and Drexel University Library and may contain materials and content, including computer code and tags, artwork, text, graphics, images, and illustrations (Material) which may be protected by copyright law. Unless otherwise noted, the Material is made available for non profit and educational purposes, such as research, teaching and private study. For these limited purposes, you may reproduce (print, download or make copies) the Material without prior permission. All copies must include any copyright notice originally included with the Material. **You must seek permission from the authors or copyright owners for all uses that are not allowed by fair use and other provisions of the U.S. Copyright Law.** The responsibility for making an independent legal assessment and securing any necessary permission rests with persons desiring to reproduce or use the Material.

Please direct questions to archives@drexel.edu

Timing-Driven Physical Design for VLSI Circuits Using Resonant Rotary Clocking

Baris Taskin Drexel University Philadelphia, PA 19104 E-mail: taskin@coe.drexel.edu John Wood MultiGiG Inc. Scotts Valley, CA 95066 E-mail: john.wood@multigig.com Ivan S. Kourtev University of Pittsburgh Pittsburgh, PA 15261 E-mail: ivan@engr.pitt.edu

Abstract—Resonant clocking technologies are next-generation clocking technologies that provide low or controllable-skew, low-jitter and multi-gigahertz frequency clock signals with low power consumption. This paper describes a collection of circuit partitioning, placement and synchronization methodologies that enables the implementation of high speed, low power circuits synchronized with the resonant rotary clocking technology. Resonant rotary clocking technology inherently supports (and requires) non-zero clock skew operation, which permits further improved circuit performances. The proposed physical design flow entails integrated circuit partitioning and placement methodologies that permit the hierarchical application of non-zero clock skew system timing. This design flow is shown to be a computationally efficient implementation method.

I. INTRODUCTION

Achieving controllable-skew, low-jitter synchronization with low power dissipation is a major milestone for digital synchronous very-large-scale integration (VLSI) circuits operating at higher frequency regimes. To reach this milestone, designers may use alternative methodologies, such as multiple clock domains or wireless [1] and transmission line-based [2–5] clocking technologies. These technologies must be supported by specific design flows and computer-aided design (CAD) suites in order to be viable in semiconductor implementation. In this paper, a physical design flow for circuits synchronized by a transmission line-based clocking technology—the resonant rotary clocking technology [5,6]—is described.

This paper is organized in follows. In Section II, a brief review of resonant clocking technologies is presented. In Section III, the proposed physical design methodology is described. In Section IV, experimental results of for various stages of the proposed physical design flow are presented. Conclusions are suggested in Section V.

II. RESONANT CLOCKING

The prevailing methodology to generate high-frequency clock signals is to use on-chip frequency multiplication with phase-locked loop (PLL) components. The on-chip PLL components occupy chip area and lead to problems with signal reflections, capacitive loading and power dissipation that effectively limit the maximum operating frequency. Also, in nanoscale complementary-metal-oxide semiconductor (CMOS) technologies, the distribution of the clock signal from a single clock source over a clock tree network [7] has become quite error-prone due to signal integrity issues. The resonant clocking technologies [2–6] present an alternative to generating the synchronizing clock signal through eliminating the necessity to use a complicated on-chip PLL component. The implementation of resonant clocking technologies requires long interconnects on the chip, which are modeled by transmission lines. Instead of the lossy *RC* characteristics of long wires, (*R*)*LC* characteristics of transmission lines provide the physical medium for oscillation. A common signal is excited and kept oscillating on transmission lines, which constitutes the global clock signal.

There are three major types of resonant clocking technologies offered to date. These resonant clocking technologies are categorized with respect to their oscillator types:

- 1) Coupled LC oscillator based clocking [2],
- 2) Standing wave oscillator based clocking [3, 4],
- 3) Traveling wave oscillator based clocking [5,6].

Coupled LC oscillator based resonant clocking technology provides a constant magnitude clock signal with a constant phase. Such properties are similar to those of a conventional clocking technology. Consequently, the main advantage of coupled LC oscillator based resonant clocking technology over other resonant clocking technologies is the minimal change to the conventional physical design flow. Higher circuit performances are achievable solely by replacing the conventional clock distribution network with that of the coupled LC oscillator based resonant clocking technology.

Standing wave oscillator based resonant clocking technology provides a varying amplitude clock signal with a constant phase. Similar to coupled LC oscillator based resonant clocking technology, the clock phase is constant, thus, this technology does not require drastic modifications to the conventional physical design flow.

Traveling wave oscillator based resonant clocking technology is the resonant clocking technology of interest in this work. Traveling wave oscillator based resonant clocking technology, also called *rotary clocking technology*, provides a clock signal which has a constant magnitude and varying phase. Varying phase (delay) of the clock signal permits easy implementation of non-zero clock skew systems. Such systems permit improved circuit performances [8].

A. Rotary (Traveling Wave Oscillator) Clock

Rotary traveling-wave oscillators (RTWO's) comprise a next-generation clock network implementation technology providing controllable-skew, low-jitter, GHz range clocking with fast transition times and low power consumption [5]. RTWO's are generated on cross-connected transmission lines, constructing a differential LC transmission line oscillator. These oscillators generate multiphase square waves with low jitter and controllable skew (360 degrees). Multiple RTWO's can be connected together forming the rotary oscillator arrays (ROA) which is the clock distribution network for the rotary clocking technology. The basic ROA structure is shown in Figure 1 ([5]). This arrangement produces a clock signal



Fig. 1. Basic rotary clock architecture.

in each ring which sweeps around the ring in a frequency dependent on the electrical length of the ring. Pulses on each ring are phase-locked via the shared transmission line wires between the rings.

Due to the ring structure of ROA's, the clock phase required to synchronize a synchronous component can be selected with fine granularity of skew [up to 360 degrees as shown in Figure 1(b)]. The clock phase driving a synchronous component is determined by the location of the connection point of the clock signal wire on the ROA ring.

The anti-parallel inverter pairs [Figure 1(c)] are used between the cross-connected lines to save power, initiate and maintain the traveling wave. After excitation, the anti-parallel inverters feed the traveling wave in the stronger direction, up to a stable oscillation frequency. The dissipated power on the ring is given by the l^2R dissipation instead of the conventional CV^2f expression. This is so because the energy that goes into charging and discharging MOS gate capacitance (of the inverters) becomes transmission line energy, which in turn is circulated in the closed electromagnetic path. The operation of the ROA structure—implemented in 2.5 V 0.25 μ m CMOS technology oscillating at various frequencies including 3.4GHz—as a new clocking technology is confirmed by simulations in [5]. Promising results of 5.5 ps clock jitter and 34-dB power supply rejection ratio (PSRR) are measured.

Two other very important metrics (for any oscillator) are the sensitivity to changes in temperature and supply voltage. It has been shown that the frequency deviation with temperature change between -50° C and 150° C is only 1% while the change with V_{DD} deviation between 1.5 and 3.5 V is around 2%. The immunity of the RTWO signals to process variations while allowing full skew control over 360 degrees of phases on the ring proves very valuable for deep sub micron applications.

B. Timing Requirements of Rotary Clocking

It is known that despite the hardship in providing multiphase, non-zero clock skew synchronization with conventional clock generation methodologies, such systems are superior compared to the traditional zero clock skew, single-phase systems in permitting higher clock frequencies and improved tolerance to process variations [8–11]. Rotary clocking technology readily supplies such multi-phase clocking with a fine grain of clock delays (non-zero clock skew).

From a CAD perspective, continuous delay models are used to model clock delays available in the network. From a circuit design perspective, the assignment of different clock delays to the synchronous components of a rotary-clock synchronized circuit are essential for a relatively uniform clock loading. In order to preserve the synchronization of an original zero clock skew circuit with rotary clocking, all synchronous components must be driven by the same location on the ROA rings. Such a load distribution may affect the rotation of the oscillatory signal on the ring, thereby causing degradations in the quality of synchronization. Also, due to simultaneous switching of synchronous components, this type of distribution might lead to thermal hot spots on the chip area. In the optimal scheduling scenario, the clock delays at the synchronous components are distributed relatively evenly in time, leading to a relatively balanced distribution of the latching points on the rotary ring. The required balanced loading of the ROA rings can be provided by clock skew scheduling [8].

Following from the discussions above, it is stated the implementation of circuits synchronized with the rotary clocking technology not only supports but also *requires* the use of nonzero clock skew, multi-phase synchronization schemes. Such circuits benefit from both the advanced timing methodologies and the rotary clocking technology.

III. DESIGN METHODOLOGY

Synchronization of digital VLSI circuits with the rotary clocking technology and the integration of non-zero clock skew, multi-phase design into the circuit design flow require methodical introduction. In this section, these new design paradigms are outlined from the physical design and electronic design automation points of view.



Fig. 2. A flow chart for the physical design flow of digital VLSI circuits synchronized with the resonant rotary clocking technology.

The proposed physical design flow is illustrated with the flow chart in Figure 2. The flow includes processing the design entry to investigate the complexity and requirements of the circuit, *partitioning* the netlist, performing *clock skew scheduling* and performing register and logic *placement*.

The implementation of the ROA rings and netlist partitioning are interdependent as illustrated in the *Partitioning* step in the flow chart. The size and number of rings in the ROA structures depend on factors such as the complexity of the design, the availability of clock network design resources, the computational resources for timing analysis, and the availability of silicon area. Despite these dependencies, the number and dimensions of ROA rings in a circuit are quite flexible. The number of ROA rings is usually held sufficiently high in order to limit the total wire length. The shapes of ROA rings are not necessarily regular (*e.g.*, rectangles) as implied by the mesh structure described in Section II-A. Such flexibility in the physical implementation of the ROA rings enables reconciliation of the non-routable blocks of the chip area.

Partitioning is performed on a gate-level or a register-toregister level netlist. For the former case, it is often necessary to insert extra registers in the logic network as part of the timing-driven partitioning process. This process is represented by the "Register Insertion" block in the flow chart. These inserted registers are level-sensitive latches operating in the transparent phases of operation in order to preserve the functionality of the original circuit. The feasibility of the partitioning result is checked at the next validation step. The *partitioning* step of the design flow is repeated as necessary.

In the *clock skew scheduling* (CSS) step, the rotary clock network is constructed. An initial timing information of the circuit is necessary for the application of clock skew scheduling. This information can be obtained by performing static timing analysis on a preliminary placement and routing or a silicon virtual prototype of the circuit. In CSS, data paths that are local to each partition are identified and the corresponding timing constraints are included in the clock skew scheduling problem for that partition. Similarly, the timing constraints of local data paths which span different partitions are included in the clock skew scheduling problem. A heuristic method is proposed to solve the partition and top block LP problems.

At the *placement* step, the optimal clock delays at each synchronous component are known. Depending on the number of clock phases and the number of registers for a given clock phase, the mapping of synchronous components to the registers within an ROA ring is performed. This is an automated design step called "Register Mapping" in the flow chart. The rest of the logic within a partition is placed in the area available within the ROA rings for this partition. This placement is performed using conventional logic placement techniques.

These three main steps of the proposed physical design flow are explained in detail in the following subsections.

A. Timing-Driven Partitioning

Traditional timing-driven partitioning methods to date are categorized as path-based and net-based partitioning, both aiming to limit the weight of cuts for circuit placement methods [12]. An alternative partitioning approach is proposed here with selection criteria that leads to partitions which are amenable to non-zero clock skew operation, each synchronized under an ROA ring. In implementation, a partitioning tool *Chaco* [13] from Sandia National Laboratories is used.

Among the criteria employed in partitioning are the weight, number and location of the cuts, the relative assignments of sequentially-adjacent registers to partitions and the number of internal vertices per partition. For the partitions to be amenable to non-zero clock skew operation, the partitions are *heuristically* enforced to be registered-input and registeredoutput systems. To ensure such a property, the fanin paths of synchronous components are assigned low edge weights. The partitioning tool minimizes the cut weights, leading the cuts to pass through the data inputs terminals of synchronous components. A synchronous component on the border of two partitions is visualized as shared between two partitions, structuring the registered-input and registered-output scheme.

The Chaco partitioning tool can be operated with different priorities assigned to multiple criteria. A balanced priority assigned between minimizing the total cut weight and increasing the number of internal vertices in all partitions is selected.



Fig. 3. Partitioning a circuit for timing analysis. The black dots represent registers and the lines represent the data paths. The data paths which are on a cut are identified and the timing analysis on these paths are performed on the higher hierarchical scale. Some paths from partition (4,1) are demonstrated.

Experimentally, this selection is observed to be sufficiently effective. The number of internal vertices being high, as opposed to having a high number of border vertices between partitions, increases the level of independence of the clock skew scheduling processes on each partition.

For RTL-level netlists, the partitioning program may validate a cut on a net that is between two combinational components. In such instances, "Register Insertion" is used to satisfy the registered-input, registered-output scheme. The number of inserted registers depends on the quality of the partitioning tool and the complexity of the design. Therefore, in designs where die area is a strict resource, the partitioning step must be applied with caution. The general partitioning process is illustrated in Figure 3.

B. Non-Zero Clock Skew Scheduling

Due to the registered-input, registered-output partitioning scheme, the clock schedules of partitions can be computed relatively independent of each other. In the proposed method, first the clock schedule of each partition is computed using a conventional clock skew scheduling method such as the method in [8,9]. Then, the clock schedule of the top block is computed in order to ensure compatibility of individual blocks. This heuristic method does not guarantee optimality, however, leads to smaller clock skew scheduling problems (per partition and top block). Incompatibilities of the results are corrected with iterations. In the iterative process, either clock skew scheduling of necessary blocks are repeated, or delay padding is used to modify the timing of one or more blocks. These iterative processes are not explained or experimented with in this paper due to the validity of the results in demonstrating the feasibility of the design flow without such iterations.

C. Timing-Driven Register Placement

In the register placement methodology, designated areas for register placement are reserved underneath the ROA rings. Highly populated register banks are stacked inside these designated regions, available for use with the full spectrum of clock phases. Upon synthesis of the circuit and the computation of optimal clock phases, each register in the synthesized netlist is physically mapped to a register underneath the ROA ring. To complete the placement step, the synthesized blocks of combinational circuitry are distributed in the free space inside the region, outside the designated areas.

IV. EXPERIMENTAL RESULTS

The development of a design tool following the guidelines of the presented design methodology is performed in C and C++ in an open source environment. Results for some of the critical individual components are presented, demonstrating the feasibility of the proposed physical design flow.

The Chaco partitioning program is tested on real and artificially generated circuits with various netlist sizes. Detailed results are reported for the ISCAS'89 benchmark circuits and for one industrial circuit industrial1. The industrial circuit industrial1 has 107875 circuit components (including 14031 synchronous components), which consist of 65908 components for logic functionality (remaining components are for test purposes). Path enumeration of industrial1 cannot be completed within the available computing resources. Thus, gate-level partitioning is performed on industrial1. The ISCAS'89 benchmark circuits are sufficiently large, yet, path enumeration can be successfully performed on these circuits. Thus, partitioning is applied to the register-to-register level netlist for these circuits. The partitioning results for

- ISCAS'89 benchmark circuits, with register-to-register level netlist partitioning,
- Industrial1 circuit, with gate level netlist partitioning,

are reported on a PowerMac computer with dual G5 1.8GHz microprocessors and 3GB RAM running Mac OS X.

In order to profile the number of inserted registers for different partition sizes, experiments are performed on industrial1. As discussed in Section III-A, the partitioning step favors the number of interval vertices to be high and number of boundary vertices to be low. Also, the set sizes must not be heavily unbalanced for proper synchronization (Section II-B). The statistics of partitioning industrial1 into grid sizes of 2x2, 4x4, 5x5, 6x6 and 10x10 by Chaco are presented in Table I. It is observed that as the number of partitions increases, the number of registers that need to be inserted on edges (for non-register input cuts) increases. For a 5x5 sized partition, the number of inserted registers (13903) approaches the original number of registers in the circuit (14031). In general, it is important to select the size of partitions (number of ROA rings) properly by considering the size of the circuit to prevent such impractical occurrences.

TABLE I INDUSTRIAL¹ PARTITIONING STATISTICS.

Partition Size	2x2	4x4	5x5	6x6	10x10
# Internal Verts	59393	49385	45616	42857	27787
# Boundary Verts	6515	16523	20292	23051	38121
# Inserted Regs	3011	9751	13903	16172	32131
Run time	15s	23s	28s	32s	54s



Fig. 4. Quality of partitions for ISCAS'89 circuits.

For ISCAS'89 suite of benchmark circuits, partitioning is performed on the register-to-register level netlist, thus, register insertion is not necessary. The clock skew scheduling problems of the partitions are solved such that optimal solutions are obtained, which lead to approximately 30% shorter clock periods on average compared to traditional, zero clock skew, edge-triggered circuits (as expected, identical to those reported in [8]). Total run time of the clock skew scheduling process is improved 28% on average on four processors, where the largest circuit (\$38417) requires 1845 secs (as opposed to 7707 secs for conventional application-76% improvement). The quality of partitions obtained for ISCAS'89 suite of benchmark circuits is shown in Figure 4. Note that in the ideal scenario, the number of internal vertices must be identical to the total number of vertices. The run times for Chaco are quite small, with the largest ISCAS'89 benchmark circuit s38417 (1636 registers and 28082 paths) taking 1.28 seconds to partition into four (4) partitions. Overall, Chaco generates partitions that are well suited for clock skew scheduling in reasonable run times.

In Figure 5, one of the ROA rings of a typical circuit designed with a 0.13 μ m technology on a 2mm x 2mm circuit die is illustrated. The die area is evenly divided into 16 regions in a four by four setting (not shown), each of which is synchronized with an ROA ring. The dimensions of each ROA ring is 500 μ m by 500 μ m. In the 0.13 μ m technology, a size of a register is considered 4μ m by 4μ m, with a minimal spacing of 2μ m between two instances. Therefore, there is enough space to place approximately 80 registers on each ROA ring edge $[(500+2)/(4+2) \approx 80]$ for a single row of registers. For 4 sides of an ROA ring and 16 rings, a total of 5120 registers are available for mapping against the synthesized logic. This number is adequate for most state-of-the-art digital circuit designs of similar die size. The dimensions of the designated area for register placement and the number of register bank rows are the determining factors for the number of registers in a design, which can be altered for particular design budget requirements. Availability of registers in the register bank enables a good distribution and mapping of clock phases to the synchronous components of a circuit.



Fig. 5. Illustration of an ROA ring in a chip layout.

V. CONCLUSIONS

In this paper, a physical design methodology for timingdriven physical design of digital VLSI circuits with resonant clocking is introduced. The methodology steps have been demonstrated to be functional and efficient, leading to both speedups in execution run time and improvements in the performance of designed circuits.

REFERENCES

- B. Floyd, X. Guo, J. Caserta, T. Dickson, C.-M. Hung, K. Kim, and K. O. Kenneth, "Wireless interconnects for clock distribution," in *Proceedings* of the ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems, December 2002.
- [2] S. C. Chan, P. J. Restle, N. K. James, and R. L. Franch, "A 4.6 ghz resonant global clock distribution network," in *Proceedings of the IEEE International Solid-State Circuits Conference*, February 2004, pp. 341– 343.
- [3] V. L. Chi, "Salphasic distribution of clock signals for synchronous systems," *IEEE Transactions on Computers*, vol. 43, no. 5, pp. 597– 602, May 1994.
- [4] F. O'Mahony, C. P. Yue, M. Horowitz, and S. Wong, "Design of a 10ghz clock distribution network using coupled standing wave oscillators," in *Proceesings of the IEEE/ACM International Design Automation Conference*, Anaheim, CA, June 2003, pp. 682–687.
- [5] 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, November 2001.
- [6] J. Wood, "Electronic circuitry," United States Patent Application Number 20030128075, July 2003.
- [7] E. G. Friedman, Clock Distribution Networks in VLSI Circuits and Systems. IEEE Press, 1995.
- [8] I. S. Kourtev and E. G. Friedman, *Timing Optimization Through Clock Skew Scheduling*. Kluwer Academic Publishers, 2000.
- J. P. Fishburn, "Clock skew optimization," *IEEE Transactions on Computers*, vol. C–39, no. 7, pp. 945–951, July 1990.
- [10] K. Ravindran, A. Kuehlmann, and E. Sentovich, "Multi-domain clock skew scheduling," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, November 2003, pp. 801–808.
- [11] B. Taskin and I. S. Kourtev, "Linearization of the timing analysis and optimization of level-sensitive digital synchronous circuits," *IEEE Transantions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 1, pp. 12–27, January 2004.
- [12] C. C. Ababei, S. Navaratnasothie, K. Bazargan, and G. Karypis, "Multiobjective circuit partitioning for cutsize and path-based delay minimization," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, November 2002, pp. 181–185.
- [13] B. Hendrickson and R. Leland, "The chaco user's guide: Version 2.0," Sandia National Laboratories, Albuquerque, NM, Tech. Rep., July 1995.