Physical Design Variation in Relative Timed Asynchronous Circuits

Tannu Sharma Kenneth S. Stevens
tannu.sharma@utah.edu, *kstevens@ece.utah.edu*
Electrical & Computer Engineering, University of Utah

***Abstract*—Variations in integrated circuits stem from multiple sources. This paper studies variations in placement and delay that occur when using commercial EDA in a relatively unsupported fashion – to implement large unclocked circuits. A tool suite is built to study placed and routed designs. Significant variations in physical placement is shown, leading to degradation in performance, power efficiency, and robustness. An experimental method of mitigating timing and placement variation using relative place directives is applied, resulting in circuits that are 7% faster and 4% lower power.**

I. Introduction

Technology scaling has had a profound effect on our ability to continually improve semiconductor designs. However, as technology scales, variations due to process, voltage, and temperature (PVT) negatively impact performance, power, and yield. Classic sources of process variations include patterning proximity effects, line-edge and line-width roughness, polish variation, and gate dielectric thickness. More recently random dopant fluctuations, implants and anneals, high-stress capping layers and gate material granularity have emerged as significant contributors to process variation. Such random and systematic variation components can effect the performance of a design [1]. Innovation to address these issues is an important ongoing process.

Another important source of variation, which is often overlooked, arises from electronic design automation (EDA) tools. Some of the variation is caused by intentional approximations in algorithms to reduce run-time of an EDA tool due to the complexity of finding optimal solutions in complex designs. EDA tool algorithms are also optimized and tuned assuming clock based designs. When applying novel design approaches, such as relative timed (RT) design, many algorithms in the commercial EDA tool flow are found wanting. This work evaluates variations that result from EDA tools when applied to relative timed asynchronous design, and an approach to reduce variations based on placement constraints. An alternative approach, not taken here, would be to develop custom asynchronous physical design tools and algorithms [2], [3].

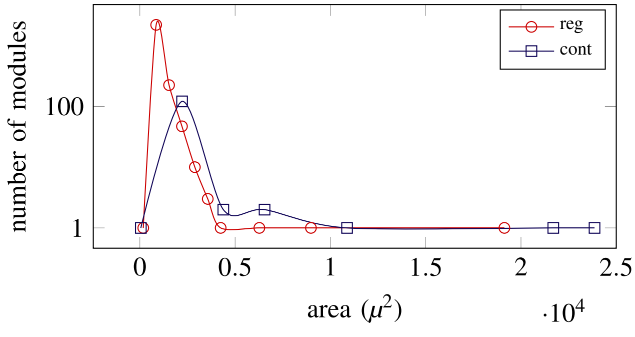


Fig. 1: Variation in area of RT elements in the encryption chip design with no placement constraints. "Cont" are the pipeline handshake controllers, and "reg" are 32-bit register banks.

Asynchronous designs have no clocks. Synchronizing signals are generated locally, derived from handshake logic. The relative timing method allows path based timing constraints to be specified [4]. Timing is specified with sdc (Synopsys design constraints). Commercial EDA tools perform timing driven synthesis and placement using these constraints. The clocked EDA tools perform well on a majority of the design elements, except a few outliers that fail due timing violations. Hence ECO repairs on final layout are required to prepare a RT design for manufacturing.

The source of timing failures in an RT design has its roots in poor physical placement. This primarily occurs when clocks are not employed and timing is exclusively specified using sdc. This EDA tool variation is illustrated with a fabricated cryptographic chip in the 65 nm node. It contains 130 linear pipeline stages using identical handshake controllers. Each pipeline stage clocks up to 24 32-bit wide register banks. Each pipeline stage has similar complexity and functionality. Identical cycle-time constraints are specified for each stage.

Fig. 1 shows area covered by standard library cells in the post layout design for each register bank and handshake controller. Fig. 2 plots the euclidean distance between pipeline

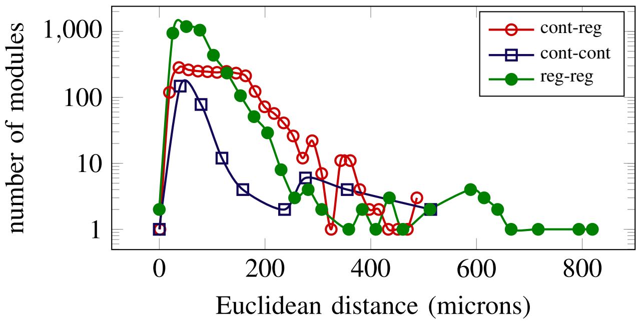


Fig. 2: Placement variation in the cryptographic chip with no placement constraints. Post layout distance between pipeline stages identified by controller to controller and register to register measurements. Controller to register distances identify pipeline stage compactness.

978-1-5090-6762-6/17/$31.00 ©2017 IEEE

stages. Module area and distances between pipeline stages would ideally be identical in this homogeneous architecture.

EDA tools perform well in placing a majority of the modules. Unfortunately there are significant outliers; some having values 10 times greater than the mean. Since the worst case condition limits the performance of a linear pipeline, these variations have profound impact on the performance and power of this design. Most if not all of these outliers also require ECO repair to meet timing.

A framework was developed in order to understand EDA tool variation in cell placement of relative timed designs and to reduce the number of manual ECOs that are required. It identifies the communicating modules at every level of hierarchy. It also measures the bounding area and center point of hierarchical module in a physically placed design which is used to calculate distance between connected modules. This enables measurement of the magnitude of physical placement variation in a design, and evaluation of the resultant impact on performance and power.

Our approach utilizes advanced placement constraints to mitigate EDA tool placement variation. Each design is divided into several hierarchical blocks (such as register banks and handshake controllers) with relative placement constraints applied. The impact of the approach in mitigating EDA tool variation, performance improvement, and power reduction is reported.

II. Background

The RT designs evaluated in this paper use bundled data paths where handshake controllers are implemented as asynchronous finite state machines [5]. Communicating handshake controllers create a silicon oscillator that dictates the frequency of operation for each pipeline stage, synchronizes between stages, and generates a local clock signal for each latch. Combinational data path logic is synthesized from behavioral specifications.

A. Relative Timing

The Relative timing methodology formally defines timing relations in an asynchronous design. An RT constraint establishes a timing relationship between two timing paths that start from a common point of divergence (pod) and end at two distinct points of convergence: $\mathsf{pod} \mapsto \mathsf{poc}_0 + m \prec \mathsf{poc}_1$ [4]. A relative timing constraint requires the maximum delay from the pod to poc_0 plus a margin (m) must be less than the minimum delay from the pod to poc_1.

Both maximum and minimum delay constraints are important for correct functioning of an asynchronous system. Relative timing constraints are presented to the EDA tools as `set_max_delay` and `set_min_delay` constraints. These sdc constraints are the only mechanism employed to identify timing paths in relative timed circuits as no clocks are present.

The intended effect of RT constraint on layout, and thus on timing, are classified into three different categories – (a) short path max-delay RT paths passing through a few standard cells (usually three or fewer) should apply strong attractive forces to tightly place those cells. These short path max-delay RT constraints are common between standard cells inside a single handshake controller module and register bank. (b) Long path max-delay relative timing constraints (RTCs) are typical for combinational functions in a pipeline stage. The attractive forces should weaken as max-delay constraints pass through more standard cells, and placement optimizations should focus more on regularly placing pipeline stages. (c) Min-delay constraints are primarily used between pipeline stages to reduce pipeline frequencies to match the combinational function of the pipeline, and optimally would be used to create regularly spaced pipeline stages. Layout optimization for min-delay constraints should be timing driven, and should apply repelling forces proportional to the delay.

Unfortunately placement of standard cells appear to be weakly constrained by `set_max_delay` in commercial EDA. Path length does not appear to play any role. The `set_min_delay` constraints are considered only for post-layout processing.

B. Relative Placement

Relative place (RP) functions in Synopsys ICC are evaluated for their ability to reduce variation in RT designs. RP constraints are applied to create the desired effect of tightly grouping standard cells that have short max-delay RTC paths.

In relative placement (RP), the arrangement of a module is defined in the form of a matrix where a cell library instance is assigned to a row and column position. The relative placement of the cells follow the exact specification of the matrix. However, other cells with fixed locations such as substrate and well taps may be interleaved in the matrix in the final placement. The relative placement functions allow an RP block to be placed in a fixed location, or allow the EDA tools to determine the placement of the block. The RP method optimizes a design for area-time if the RP groups are judiciously selected [6].

In this paper, all standard cells in a module are not RP constrained. Only cells that have short path max-delay constraints are placed in RP groups. This allows a module to span a larger layout area, because gates such as an inverter on an input are unconstrained and can be placed at significant distance from the rest of the module.

C. Relative Timed Physical Design Flow

The RT design flow, in Fig. 3, is built upon the traditional synchronous CAD tools with some additional steps: (a) Design and characterize RT modules [7]. (b) Map RTC and timing margins onto design instances in an architecture [8]. (c) Timing close using the supplied RTC delay targets. (d) Perform post-layout RTC timing validation [9]. This entire flow allows commercial EDA tools to be used to design, synthesize, perform physical design, and validate RT asynchronous designs.

D. Definitions

Terms *modules* and *blocks* can be used interchangeably during the course of this paper. They are reusable standard

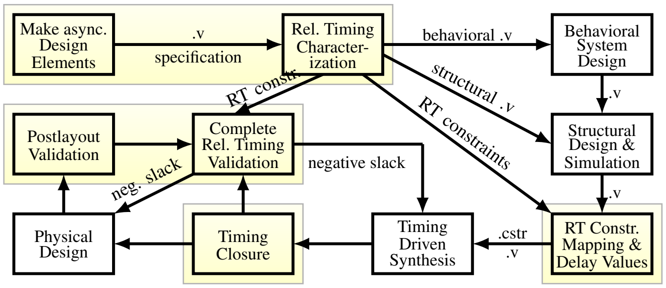


Fig. 3: Simplified Relative Timing Design Flow

blocks composed of standard library cells. RP constraints are primarily applied to register banks and handshake controllers. These blocks can be present at any level of hierarchy. The timing of a module and its connected blocks are defined using relative timing constraints. *Variation* is a measurement of differences between post-layout timing, area, and distances between modules that have identical relative timing constraints.

III. Variation Evaluation

A tool is implemented to analyze EDA caused design variations as related to hierarchical modules. It extracts physical parameters of each cell from a placed and routed design. Later this database is used to generate physical parameters of modules requested by the user. It will also provide distance between specified modules.

This paper reports on the tool applied to three design examples, one of which was fabricated. For these designs, we report physical area for every controller (cont) and connected register (reg) instance, and the distance between communicating cont-to-cont, cont-to-reg and reg-to-reg pairs.

A. Evaluation

To study the amount of variation in a large design with n module instances, the gap between average (μ), majority ($2\sigma+\mu$) and maximum values is studied. Results are plotted as histograms where Rice's rule (bin $= \lceil 2n^{1/3} \rceil$) is used to calculate the number of bins and class-size, for a class-frequency distribution. For a large design, majority value is defined by two standard deviation away from the mean which captures 95% cases of all register and controller modules if the distribution is normal. The sample variance, average (μ), and 2σ values are calculated for each set to obtain two standard deviation. For a small design, the sample set is small enough to compare individual values of each module.

B. Variation Mitigation Approach

The variation mitigation approach in this paper is to apply RP constraints to force the EDA tools to effectively treat short and long path max-delay relative timing constraints differently. Modules with short path max-delay RTC have the constrained library cells placed in RP groups to enforce tighter cell placement. This consists of register banks, handshake controllers, and clock generating modules. The matrix arrangement of cells for RP is driven by RT constraints and cell connectivity.

The benefits of artificially producing two different sets of max-delay constraint layout behavior through RP commands is evaluated. This is shown to help overcome some of the weaknesses in using these tools to create asynchronous designs with reduced EDA tool variation.

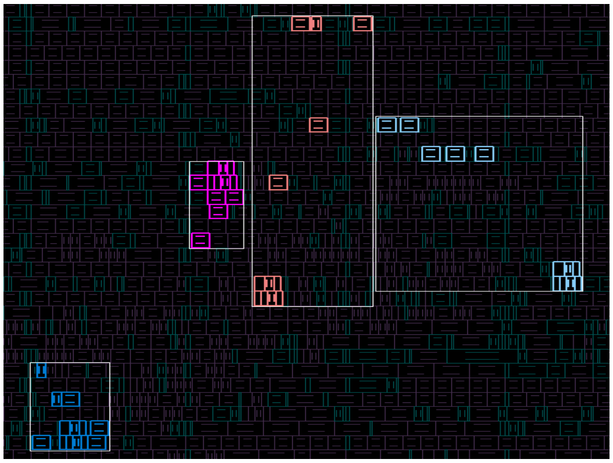


Fig. 4: Variation in the placement of controller cells at level 4 in the final layout of cryptographic chip with RPC on *all* identified sub-modules.

Many RP groups make it more difficult for the place and route tools as they create large regions where unconstrained cells can not be placed. Thus in this work we experiment with incrementally applying RP constraints to different handshake controllers and register banks in each of the three designs to also evaluate this effect.

IV. Designs

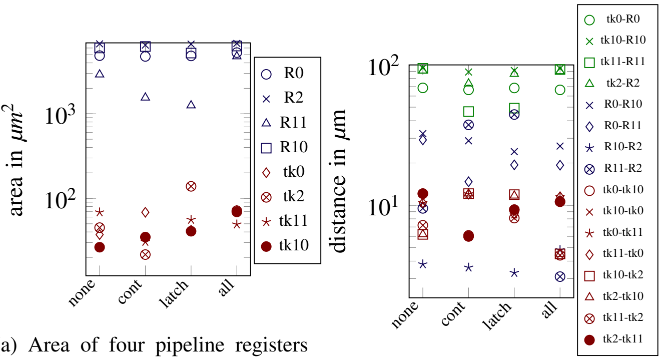
Three RT asynchronous designs are evaluated: a very simple four-stage multiplier, a 64-point fully pipelined multi-rate FFT design with lots of pipeline fan-out and fan-in with multiple frequencies, and a large cryptographic chip with a 130 stage homogeneous computation pipeline. There is substantial diversity in size and pipeline structures of these circuits, making them good examples to evaluate quality of results when industry standard EDA flows are applied to asynchronous designs.

A. Four Stage Multiplier

A simple 32-bit four-stage multiplier design implementing a square function is evaluated [10]. This small design forks into two asymmetric parallel channels. The tk0→tk10 controller channel contains a substantially more complex function than the tk0→tk11 channel. A smaller combinational function joins the two channels. RP groups are applied to the register modules and handshake control modules.

B. 64 Point Pipelined Fast Fourier Transform

This design is a 32-bit 64-point multi-rate FFT design that is hierarchically decomposed at top level to operate at four different frequencies. The frequencies are based on FFT decomposition where at the top level $N_1 = 16$ points and $N_2 = 4$ points. The 16-point FFTs are decomposed into four point FFTs ($N_1, N_2 = 4$) [11]. This multi-frequency, low power pipelined FFT has approximately 900 pipeline stages, 800



(a) Area of four pipeline registers and controllers.

(b) Distance between connected pipeline stages and controllers.

Fig. 5: Spread of area and distance across pipeline stages in a pipelined multiplier design.

latches with 3,340 cont-to-cont, 1,666 cont-to-reg, and 3,035 reg-to-reg paths. This design is expected to show variation across the placement of modules operating at different frequency at points where there is large fan-in and fan-out between pipeline stages. Three modules *(controller, latches, and PCM)* are relatively placed to evaluate EDA tools to reduce variation.

C. A Cryptographic Chip

This chip contains a 130 stage deep linear pipeline. Each pipeline stage has a similar functionality and is given exactly the same cycle time target. The 20 M transistor cryptographic chip contains four copies of a synthesized and placed block that consists of 5 M transistors each. Any variation in delay directly reduces the performance. Ideally, all stages would be equally spaced on the design. The design has roughly 2,500 32-bit registers with about 2,500 cont-reg paths, 257 cont-cont paths, and 4,064 reg-reg timing critical paths. The design uses ripple carry logic to reduce area and power at the cost of performance. The controllers, latches, flip-flops, pulse clocking module (PCM), and a godone state machine are selected for relative placement in the design.

The design was fabricated using RP on all the identified modules to reduce variation. An ECO flow was performed to repair the remaining outliers after the variation mitigation approach in this paper was applied. The chip was fabricated in the 65 nm node and is 100% functional. Average chip operating frequency is 241 MHz, with 0.40 μm as the smallest distance between communicating stages.

V. Results

Physical design is completed with Synopsys ICC. Power, performance and timing are extracted using ModelSim and PrimeTime. Parasitic extracted delays and capacitances from the layout are used in the simulations. Power numbers are generated from a vcd file that identifies node activities from the test benches. No ECO flows were applied, so the designs fail simulation if the outliers are so egregious that timing is not met in the simulations. In all design cases, 95–98% of the measured elements are within the majority value leaving only 5% of the elements as outliers. The variation in these outliers on large designs can cause timing violation or design failure.

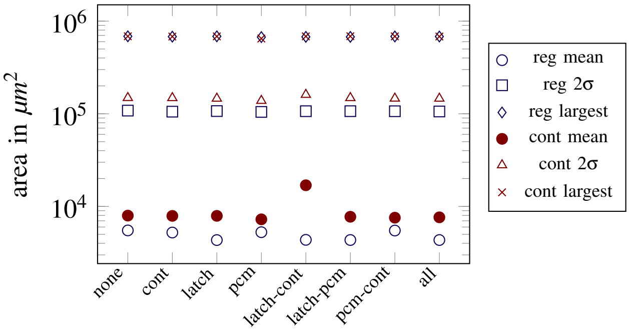


Fig. 6: Average, majority and largest area for pipeline registers and controllers in FFT-64 design with different RP groups.

The variation in the placement of a macro block on a highly structured linear pipeline design can be seen in Fig. 4. The four controller modules on same level of hierarchy have four different areas to them.

A. Four Stage Multiplier

Relative place directives are demonstrated on this example to be highly effective on small designs with irregular pipeline delays. Fig. 5 shows that adding constraints clearly reduces variation, tightly grouping short-path timing constraints while also creating more regular and optimized spacing between pipeline stages with larger RT constraint paths delays. With RPC applied to all modules, the tk0→tk10→tk2 path placement is shorter than the tk0→tk11→tk2 path. Since the former path is the critical path in this design, and the latter has ample margin, this placement is by far the best of the four. Without the RP directives, even in this small design, the critical path is less well constrained.

B. FFT-64

The multi-rate FFT design has large differences in pipeline frequency targets and logic per pipeline stage. Eight different implementations were evaluated based on applying RP constraints to different modules. Design variation is reported for area and pipeline distances.

1) Area: Fig. 6 shows average, 2σ, and maximum area value of pipeline registers and controllers on a log scale. The graphs for each implementation look much like that plotted in Fig. 1, where most of the designs well optimized with a very few significant outliers.

These graphs show that in large designs RP does not remove the need to perform ECO for a few poorly optimized modules. The difference between the 2σ (majority) and max value, the mean and max, and the min and max values are 4.3×, 20×, and 10^4× respectively.

The smallest controller area is obtained when only the pulse clock module is used. This module interfaces between adjacent pipeline stages and the controller and its local register bank. However, this configuration has a much worse average register area. The general observation is that that applying RPC to all – registers, controllers, and their related modules – provides the best results.

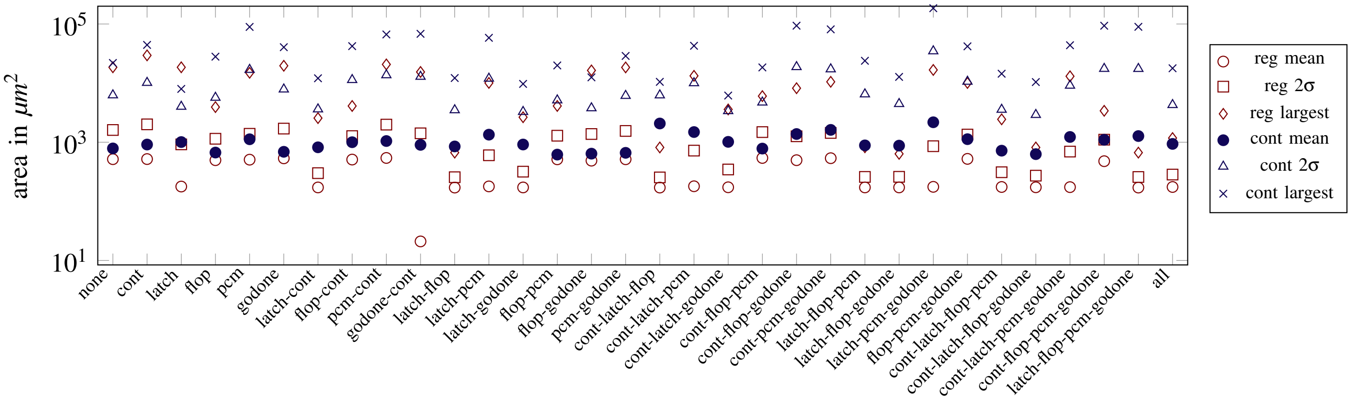


Fig. 7: The area of controllers and registers in Cryptographic chip design with different RP configurations.

2) Distance: Employing RPC to all modules with short path RTCs provides the best solution if the outliers can be fixed with ECOs. The average variation reduces by 13% between no RPC and all RPC.

Variation between the controllers and the register banks they clock (cont-reg in Fig. 8) are critical because these variations produce performance penalties. Distance between pipeline stage controllers (cont-cont in Fig. 8) and data paths (reg-reg in Fig. 8) are difficult to analyze with aggregate data due to diversity in pipeline fan-out and logic complexity in this design. Variation is largely uniform in this architecture with the 2σ value being approximately five times greater than the mean, and the largest outlier approximately four times greater than the 2σ value.

3) Timing, Power and Performance: Table I shows timing, performance, power, and energy results. Relative place constraints provided a small overall improvement on this design when ECOs are not applied to mitigate the effect of outliers. Employing all RP constraints produced the best design. It was the most efficient design as both energy per FFT and performance improves by 3% with less timing penalty. The PCM design has zero nets with DRC violations and is the second best configuration. Outliers cause test bench failures in two of the eight configurations.

C. Cryptographic Chip

RP constraints are applied to five different modules and 32 different configurations are compared. Unfortunately significant EDA tool variation clearly exists, as handshake controller area varies from 52 to 23,860 μm^2 where the 2σ value is 2,216 μm^2 as shown in Fig. 1. Perhaps the worse effect is the variation in distance between pipeline stages (Fig. 2).

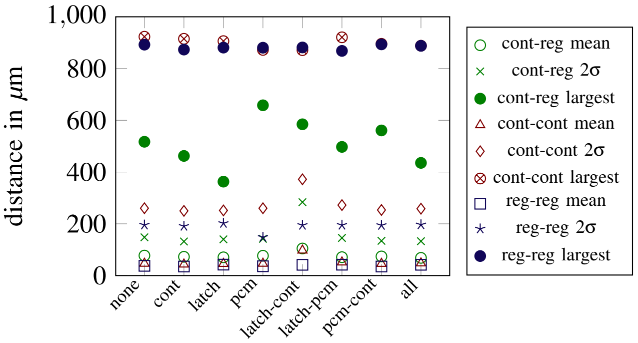


Fig. 8: Euclidean distance between connected modules for FFT-64 design with different RP groups

TABLE I: Timing and performance of FFT-64 in different designs with and without RP constraints.

Design	WNS (ps)	Violating Paths	Avg. Cycle Time (ns)	Power (mW)	Energy (nJ)	$e\tau^2$
No RPC	70	7267	1.49	15.4	22.95	55.95
RPC on all	40	7752	1.44	15.5	22.32	46.28
RPC on PCM	60	7347	1.47	15.6	22.93	49.55

1) Area: The EDA tools struggle to create high quality physical design for large homogeneous relative timed circuits. Applying RPC to all targeted modules reduce average register area three-fold over the unconstrained design (Fig. 7). Average controller area for the fully constrained design (all) is 19% larger than the unconstrained case (none). However, the 2σ and worst case area is 68% and 81% of the unconstrained values. EDA tools do a better job of optimizing area values using a subset of the constrained modules. This is likely due to reduced placement options as RP modules become a significant percent of the design layout.

2) Distance: Distance between controllers and registers and between pipeline stages ideally have very little variation in this homogeneous design. Unconstrained (none) controller to register distances (cont-reg in Fig. 9) between the mean and the 2σ and worst values vary by 2.3× and 4.6×. The fully constrained case (all) has one of the better ratios, and improves the variation to 2.1× and 2.4× respectively. The interesting latch-pcm-godone and latch-flop-pcm-godone case has by far the largest delays.

The quality of placement is shown by the distance between handshake controllers (cont-cont in Fig. 9) that communicate between pipeline stages. The fully constrained (all) case creates some of the smallest distances. The controller mean placement is improved 1.9× over the unconstrained design (none). However, the difference between the mean value and 2σ and max values are 4.0× and 13.7× for the fully constrained design.

The data path (reg-reg in Fig. 9) is better optimized by the EDA tools. This is likely because min-delay optimizations, that are used for critical delay matching of the control path are largely considered as post-layout ECO. The fully RP constrained (all) mean value for the data path is only 6%

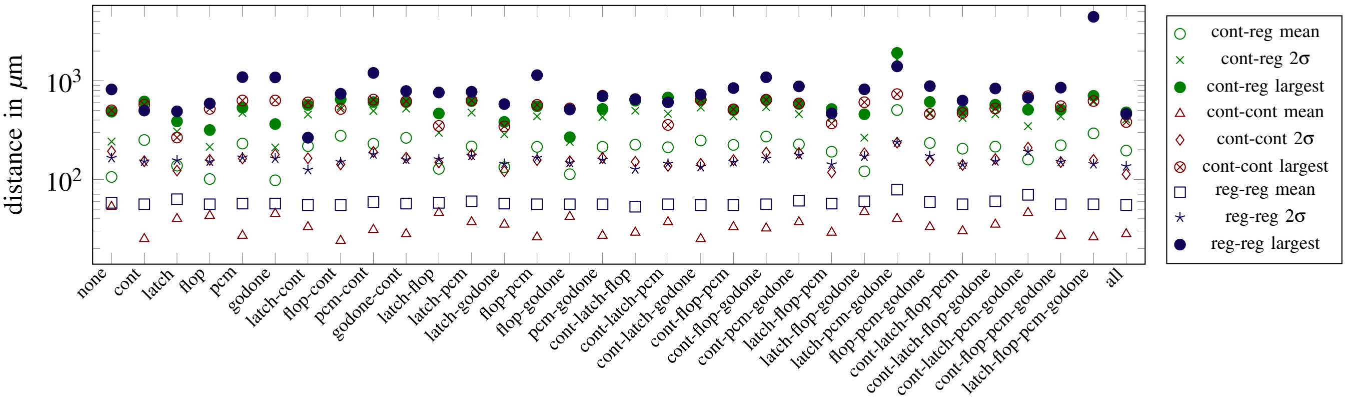


Fig. 9: Euclidean distance between connected modules in Encrypted-chip design

better than the unconstrained case (none). However, the 2σ and max values show a 1.2× and 1.8× improvement. In a linear pipeline it is the worst case values that dictate performance.

3) Timing, Power and Performance Results: Timing, performance, power, and energy per operation with and without RPC are shown in Table II. No ECOs were applied to fix outliers in these results. The design latch-pcm-godone failed the test bench due to outliers.

Best case improvements in the 32 designs reduced energy by 8%, improved $e\tau^2$ by 18% (cont-flop-godone), and increased performance by 7% (pcm-cont). Applying all constraints results in a design that is 5% faster, but also has a 5% energy penalty, resulting in a 5% improvement in $e\tau^2$. This design has five nets with DRC violation compared to 159 violating nets with no RPC applied. The cont-flop-godone configuration has the fewest number of paths that failed timing and fewest DRC violations (2).

VI. Conclusion

This paper evaluates a method to enforce different placement optimizations on short and long path maximum delay relative timing (RT) constraints through the application of relative place (RP) constraints. RP groups are formed to tightly place standard cell gates with short path max-delay RT constraints. However, the variation mitigation approach does not eliminate the need to run ECOs to meet min-delay RTC.

Poor EDA tool placement of a number of pipeline cells produces variations that result in a significant number of paths that violate timing constraints and reduce performance, as well as increase power. This work reduces variation when commercial EDA tools are used in physical design of RT asynchronous circuits.

The evaluation tools developed here are applied to three designs that span a large range of complexity and pipeline structures. The largest design was fabricated and is fully functional after applying ECO to fix variation outliers. Numerous versions of these three designs were fully implemented to evaluate the benefit of applying RP constraints on different sets of short path max-delay constraints. The designs were evaluated for area variation, pipeline stage distance variation, worst negative slack, violating paths, performance, and energy.

The work shows that applying RP constraints to short path max delay constraints can significantly reduce variation, which also reduces the number of nets that need to be modified in the post-layout ECO flows. It also shows that the EDA tools struggle to optimize large clockless designs.

TABLE II: Timing and performance of cryptographic-chip in different designs with and without RP constraints.

Design	WNS (ps)	Violating Paths	Avg. Cycle Time (ns)	Power (mW)	Energy (nJ)	$e\tau^2$
No RPC	240	280	2.467	474.1	1.17	7.12
RPC on all	130	290	2.350	521.4	1.23	6.79
RPC on cont-flop-godone	110	80	2.365	457.0	1.08	6.04
RPC on pcm-cont	280	4309	2.306	494.0	1.14	6.06

References

[1] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, and S. Mudanai, "Process Technology Variation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2197–2208, Aug 2011.

[2] G. Wu, T. Lin, H.-H. Huang, C. Chu, and P. A. Beerel, "Asynchronous Circuit Placement by Lagrangian Relaxation," in *International Converence on Computer-Aided Design*. IEEE/ACM, 2014, pp. 641–646.

[3] R. Karmazin, S. Longfield, C. T. O. Otero, and R. Manohar, "Timing Driven Placement for Quasi Delay-Insensitive Circuits," in *21st International Symposium on Asynchronous Circuits and Systems*. IEEE, May 2015, pp. 45–52.

[4] K. S. Stevens, R. Ginosar, and S. Rotem, "Relative Timing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 1, no. 11, pp. 129–140, Feb. 2003.

[5] I. E. Sutherland, "Micropipelines," *Communications of the ACM*, vol. 32, no. 6, pp. 720–738, June 1989.

[6] A. A. Farooqui, V. G. Oklobdzija, and S. M. Sait, "Area-Time Optimal Adder with Relative Placement Generator," in *Proceedings of the International Symposium on Circuits and Systems*, vol. V, 2003, pp. 141–144.

[7] Y. Xu and K. S. Stevens, "Automatic Synthesis of Computation Interference Constraints for Relative Timing," in *26th International Conference on Computer Design*. IEEE, Oct. 2009, pp. 16–22.

[8] E. Quist and P. Beerel, "Automated Path Specification for Static Timing Analysis of Relative Timing Designs," in *International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU Workshop)*. ACM, March 2010.

[9] W. Lee, T. Sharma, and K. S. Stevens, "Path Based Timing Validation for Timed Asynchronous Design," in *The 29th International Conference on VLSI Design (VLSID)*. IEEE, Jan 2016, pp. 511–516.

[10] K. S. Stevens, Y. Xu, and V. Vij, "Characterization of Asynchronous Templates for Integration into Clocked CAD Flows," in *15th International Symposium on Asynchronous Circuits and Systems*. IEEE, May 2009, pp. 151–161.

[11] W. Lee, V. S. Vij, A. R. Thatcher, and K. S. Stevens, "Design of Low Energy, High Performance Synchronous and Asynchronous 64-Point FFT," in *Design, Automation and Test in Europe (DATE)*. IEEE, Mar 2013, pp. 242–247.