Relative Placement in Timed Asynchronous Design

Tannu Sharma William Lee Kenneth S. Stevens
Electrical & Computer Engineering, University of Utah
Email: *tannu.sharma@utah.edu, william.lee@utah.edu, kstevens@ece.utah.edu*

***Abstract*—Timed asynchronous circuits can be implemented using commercial computer-aided design (CAD) tools. Relative timing methodology is applied for interpreting the complex timing of asynchronous circuits into CAD tools that can be understood with minimum and maximum timing constraints. Typical synchronous placement algorithm neglect timing information which is critical for asynchronous methodology. Relative timing constraints are employed during placement of the modules with the help of relative placement methodology. This paper explicitly adopts relative placement, supported by commercial CAD tools to optimize a design for its area, wire-length, distance, power, and performance.**

I. Introduction

Future generation circuits are expected to be faster than the current circuit designs. The target is to simplify the future designs or their implementation so as to achieve better performance and power benefits over the same or lesser area. Asynchronous circuit design methodologies provide power and performance benefits over the synchronous design methodologies with comparable area and wire-length.

Wire scaling does not follow Moore's law. Wire-length has been a bottleneck in achieving a better, faster and error-free design. Hence, while exploring for newer design methodologies a constant innovation in CAD is required to handle the area and wire-length challenges. A high level of design planning and route estimation is needed to support double patterning and multiple timing corners, and to enhance design productivity. Due to limited or no support for asynchronous circuit designs, additional supports in the existing commercial electronic design automation (EDA) tool flow have been done to design a circuit with asynchronous methodology [1].

Also, timing-closure between synthesis and physical design has always been a time consuming step in the design cycle, which is a challenge in an asynchronous circuit design due to its complex timing. Even though timing constraints are met during the synthesis, obeying them during the physical design stage is a task. Not all EDA tools can handle the complexity of asynchronous timing constraints during the physical design stage. This work uses Synopsys IC Compiler over Cadence Encounter, as the former can handle both minimum and maximum constraints during the physical design of a circuit, when the latter can only consider the maximum timing constraints.

In this paper, we would like to explore the benefits obtained using one such physical design methodology, relative placement, available in existing commercial EDA tools to optimize an asynchronous circuit design for power, performance, area and wire-length. The next section will discuss a background followed by the relative placement technique in section III with results and conclusion in section IV and section V, respectively.

II. Background

The basic difference in the implementation of an asynchronous and synchronous circuit design is the timing, that defines the order of sequence of events. For synchronous systems, timing is defined with a clock signal, where the cycle time must be greater than the combinational propagation delay between pipeline stages. Whereas in bundled data asynchronous systems, timing is defined for each asynchronous controller with respect to each connected register while considering the occurrence of the event in the previous controller.

Until the recent past, asynchronous circuits weren't the focus of the design industry. The complex circuit timing, presence of cyclic paths and no support from the commercial CAD tools has always prevented designers from using them. However, with decreasing feature size and competitive design requirements, they have gauged interest of the designers, as asynchronous design practices can help to achieve better power, and performance [2].

A. Asynchronous Designs

An asynchronous design consists of handshake signals, that identify the data validity and ability to accept new transactions over the communication channel. The *handshake channel* is the communication link composed of data wires, request signal (*req*) to identify data validity and acknowledgement signal (*ack*) to confirm data transaction [3].

Bundled data is one among many asynchronous design styles that has timing constraints. The linear controller (LC) blocks in Fig. 2, implement a silicon oscillator that controls the frequency of operation for each pipeline stage, synchronization between pipeline stages and also the local clock signal generation for each pipeline latch (L). This is implemented as an asynchronous finite state machine.

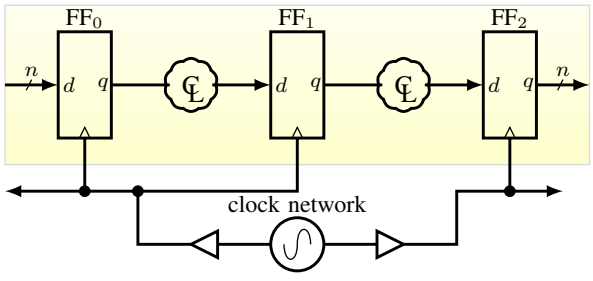

Fig. 1. Clocked design. Frequency and datapath delay of first pipeline stage is constrained by $FF_i/clk\uparrow_j \mapsto FF_{i+1}/d + margin \prec FF_{i+1}/clk\uparrow_{j+1}$

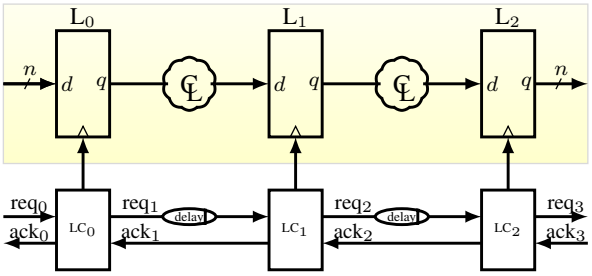

Fig. 2. Timed (bundled data) handshake design. Delay sized by RT constraint $req_i\uparrow \mapsto L_{i+1}/d + margin \prec L_{i+1}/clk\uparrow$. Each $req_i\uparrow$ handshake on LC_i indicates new data is presented to pin d of L_i.

B. Relative Timing

Relative timing (RT) is a methodology to define the timing relation within an asynchronous design that can be interpreted within a commercial CAD tool. It establishes a timing relation between the two timing paths that starts from a common point of divergence (pod) and ends at two distinct point of convergence (poc) $\mathsf{pod} \mapsto \mathsf{poc}_0 + m \prec \mathsf{poc}_1$ [4].

The methodology states that the maximum delay (max_delay) from a pod to poc_0 plus a margin (m) must be less than the minimum delay (min_delay) from the pod to poc_1. Using this methodology, relative timing constraints for an asynchronous circuit in Fig. 2 can be expressed similar to a synchronous design in Fig. 1. For example, Fig. 2 shows a simple burst-mode linear controller pipeline, establishing the timing constraints between the control path and the data path, where, data L_{i+1}/d must arrive before the clock L_{i+1}/clk for the latch to work correctly. RT timing model is used throughout our approach because of its generality across timed asynchronous designs.

C. Physical Design

Physical design is a design stage where a logical netlist is used to perform physical layout of the design in various steps starting from design planning, physical synthesis, clock tree-synthesis, placement, routing and ending at chip finish without violating any timing constraints.

D. Design Planning

Design planning is floor-planning, which is a process of gate sizing, placing cells and blocks for effective physical design. In this step, timing budgets are allocated iteratively on each block until optimal floor-plan is obtained. This step ensures timing-closure, decides placement of the blocks to have shorter critical paths, plans clock tree path, prevents routing congestion and also minimizes the routing area. In addition, it is an important step to reduce IR drop, and electro-migration problems [5].

In this step, a floor-plan is created to determine the size of a design, the design boundary and the size of core area. It also creates site rows for the placement of standard cells, I/O pads and performing power planning.

E. Placement

After design planning, Cell placement and placement optimization is performed, where it addresses and resolves timing-closure. The iterative placement is performed to generate legalized placement for leaf cells and an optimized design.

Then, the clock tree synthesis & optimization, and the routing & post routing optimization is preformed to finish the chip for manufacturing. At every step, design is optimized while ensuring there are no timing constraint violations or design rule violations.

Design planning and placement are important steps in optimizing a design. Academia has always worked on developing physical placement tools to meet the specific needs of optimizing an asynchronous circuit design [6], [7]. However, a solution within standard commercial CAD tools will drive the acceptance of asynchronous methodology in the industry.

III. Relative Placement

Asynchronous design has a complex timing and optimizing it without considering timing will give sub-optimal results. Relative placement methodology can be utilized to make use of relative timing constraints added on the design. Relative placement is performed based on the specified relative placement constraints which are applied on a gate-level netlist and verified for violations before actually placing the constrained cells at the specified locations.

The technique was originally defined for data-paths and register blocks. However, in our experiment, we constrained the combinational blocks, registers/latches, asynchronous controller modules, pulse clocking module to observe design performance and benefits.

Relative placement constraints are defined to control the placement of certain cells and/or modules connected to specific module instances so as to ensure they are placed in close proximity. A relative placement group is formed that has relative placement constraints for all the instances of a module. The defined relative placement constraints are annotated to generate a matrix structure of the instances to control its placement. The tool preserves the structure and verifies it for violations. It places the filler and connected cells automatically in the void locations [8].

Scripts have been created using tcl/tk, perl and python to observe the locations of each module instance and its cells with and without relative placement. There are scripts to automate the adoption of relative timing constraints during relative placement of a design. A database is created for the physical design data of each cell of a controller module. A set of bounding box values is generated from the cell location in a controller. These values are evaluated to calculate the bounding box of a module based on the smallest lower left coordinate in the set and largest upper right coordinate in the set. The bounding box is later used to calculate the center of each controller module.

After generating the physical data for each controller module, the values are used to calculate the area of each module and the manhattan distance and euclidean distance of the connected modules. The distance gives an approximate wire-length between connected controllers.

Also, scripts are used to extract the design performance, and wire-length data with and without relative placement. The power benefits are observed using Synopsys PrimeTime for each design with and without relative placement constraints.

In addition, the timing constraints are verified for violations for each asynchronous controller and will be used to further optimize the design for its timing [9].

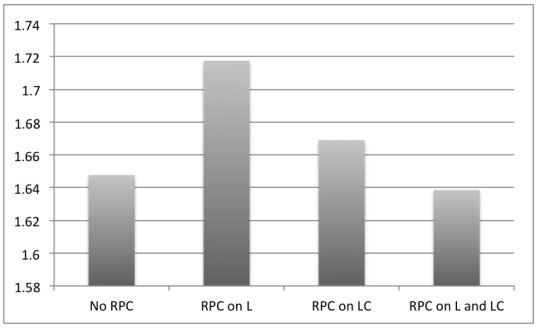

Fig. 3. Power values (mW) for multiplier design with and without relative placement constraints (RPC).

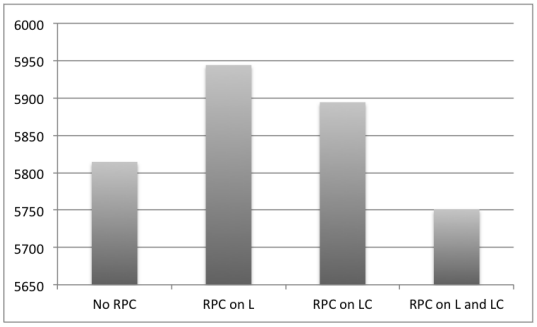

Fig. 4. Area values (μm^2) of multiplier design with and without relative placement constraints (RPC).

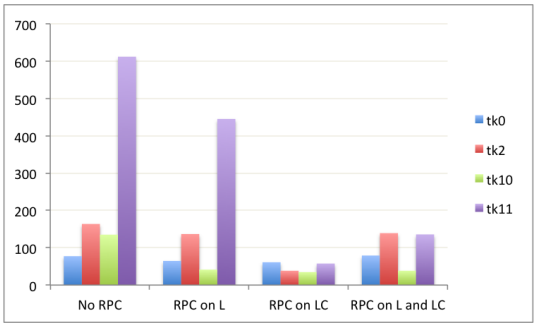

Fig. 5. Area (μm^2) of each controller in the multiplier design with and without relative placement constraints (RPC).

IV. Experimental Results

The relative placement results are evaluated for two asynchronous circuits, a multiplier and an encrypted chip design.

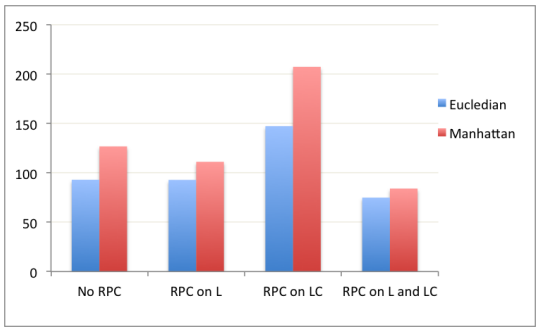

Fig. 6. Euclidean and Manhattan distance (mm) for each connected in the multiplier design with and without relative placement constraints (RPC).

Both designs uses a single type of controller instance. The multiplier design has four pipeline stages of controller that are studied for power and performance benefits from relative placement. The area and wire-length results obtained from relative placement shows significant improvement over a non-relatively placed design.

An encrypted chip is an asynchronous design with over 3 million transistors and a few hundred timing constraints. The design is a good example to observe the effects of relative placement on an industrial asynchronous design in terms of power, performance, area and wire-length with various configurations of relative placement constraints on design modules.

Both the designs have been studied with various combinations of relative placement constraints (RPC). Multiplier design is studied without RPC, with RPC on latch modules, with RPC on controller modules and with RPC on both controller and latch modules. Fig. 3 shows the power distribution on multiplier design with and without RPC. The graph clearly shows that the results obtained with RP constraints on both controller and latch modules are better than with no RP constraints. Similar results are obtained for area distribution for the same circuit with and without RPC, as shown in Fig. 4.

Fig. 5 shows the area of each controller in the multiplier design and Fig. 6 shows the Euclidean distance and the Manhattan distance between the connected controllers. Manhattan distance gives a closest approximate value of wire-length between connected modules. The graph confirms that a greater area benefit is obtained by constraining the controllers alone, however, the overall wire-length of the connected modules is also increased. Therefore, the best results obtained are when the relative placement constraints are added for both controller and latch modules.

A detailed timing analysis for each asynchronous controller is done using a path based timing validation flow [9]. The cycle time results obtained for connected controllers, with different configurations of relative placement constraints, in a multiplier design is shown in Fig. 7. The design with the defined relative placement constraints has least cycle time.

The power results for the encrypted chip design are shown in Fig. 8. The power benefits aren't significant, however, for cycle-time values referred in Fig. 9, the energy values with

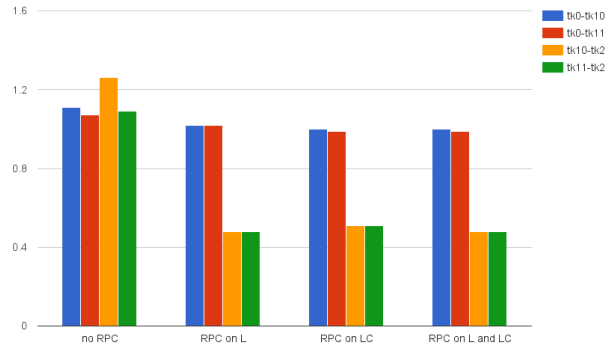

Fig. 7. Cycle-time (ns) between connected in the multiplier design with and without relative placement constraints (RPC).

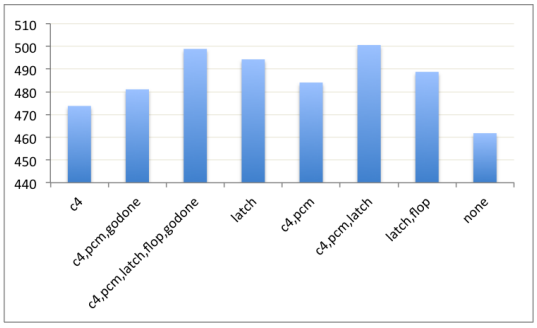

Fig. 8. Power (mW) for various configurations of encrypted chip design with and without relative placement constraints (RPC).

RPC are same as without RPC as shown in Fig. 10. The graph shows the energy distribution on the encrypted chip design with various combinations of relative placement constraints.

The results for controller area and controller connectivity are only studied with a few combinations. When RP constraints were added on controllers and pulse clocking module (PCM), the area for 129 controllers in the design was in the range of 28.8 to 21489 (micro m sq.). When RP constraints were defined for all the modules (like controller, PCM, flop, go-

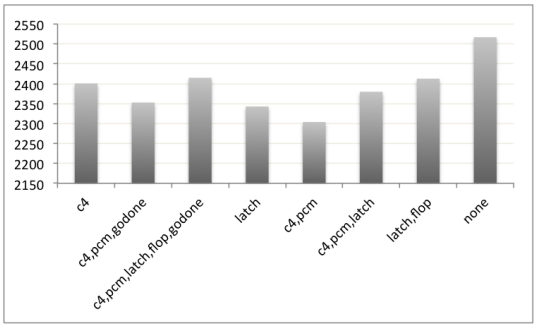

Fig. 9. Cycle-time (ps) for various configurations of encrypted chip design with and without relative placement constraints (RPC).

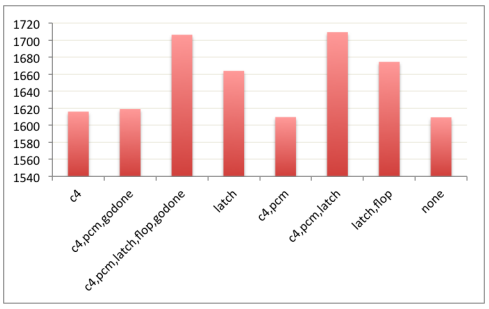

Fig. 10. Energy (fJ) for various configurations of encrypted chip design with and without relative placement constraints (RPC).

done) the area for each controller was in the range of 20.6 to 16904.5. Without RP constraints, area of the controller modules and the design increases significantly.

V. Conclusion

The physical design solution we are exploring is with the commercial EDA tools which provides an advantage of implementing the flow on any production design and evaluate the results on its power, performance, and size. It can be concluded from the discussed designs, one of them being a fairly complex design with over 3 million transistors, that the relative placement methodology can be used to optimize an asynchronous system for power, performance, area and wire-length. The results are promising and the methodology can be used during the placement of any existing design with its asynchronous design version.

References

[1] V. S. Vij, "Algorithms and methodology to design asynchronous circuits using synchronous CAD tools and flows," Ph.D. dissertation, University of Utah, May 2013.

[2] C. J. Myers, *Asynchronous Circuit Design*. J. Wiley, 1999.

[3] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design – A Systems Perspective*. Kluwer Academic Publishers, 2001.

[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] J. Y. Sayah, R. Gupta, D. D. Sherlekar, P. S. Honsinger, J. M. Apte, S. W. Bollinger, H. H. Chen, S. DasGupta, E. P. Hsieh, A. D. Huber *et al.*, "Design planning for high-performance asics," *IBM Journal of Research and Development*, vol. 40, no. 4, pp. 431–452, 1996.

[6] G. Wu, T. Lin, H.-H. Huang, C. Chu, and P. A. Beerel, "Asynchronous circuit placement by lagrangian relaxation," in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2014, pp. 641–646.

[7] E. Kounalakis and C. P. Sotiriou, "Cplace: A constructive placer for synchronous and asynchronous circuits," in *Asynchronous Circuits and Systems (ASYNC), 2011 17th IEEE International Symposium on*. IEEE, 2011, pp. 22–29.

[8] A. Farooqui, V. G. Oklobdzija, S. M. Sait *et al.*, "Area-time optimal adder with relative placement generator," in *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, vol. 5. IEEE, 2003, pp. V–141.

[9] W. Lee, T. Sharma, and K. S. Stevens, "Path based timing validation for timed asynchronous design," in *VLSI Design (VLSID), 2016 25th International Conference on*, Jan 2016.