What is Physical Synthesis?

Zhuo Li and Charles J. Alpert, IBM

VLSI technology scaling has caused interconnect delay to increasingly dominate the overall chip performance. A design that satisfies timing constraints after logic synthesis will not necessarily meet timing constraints after place-and- route due to wire delays. Physical synthesis has been emerged as a necessary weapon for design closure. It is a core component of modern VLSI design methodologies for ASIC, game chips and high performance microprocessors.

Physical synthesis begins with a mapped netlist generated by logic synthesis. The netlist describes the logical connections among the physical components (logic gates, macro/IP blocks, I/O pins, etc.). Physical synthesis generates a new optimized netlist and a corresponding layout. Its objectives are to satisfy a combination of timing, area, power, and routability. One can think of physical synthesis as a wrapper around traditional place and route, whereby synthesis-based optimization are interwoven with placement and routing.

For example, physical synthesis commonly starts by performing placement, followed by timing analysis [1]. Not surprisingly, timing analysis will generally highlight severe timing and electrical problems due to long wires. Timing optimization like buffering, gate sizing, Vt swapping, cloning, fan-in tree optimization, logic decomposition, connection reordering, etc. can then be applied to drive towards timing closure [2]. In addition, scan chain generation and clock insertion can also be run during this process.

Physical synthesis naturally proceeds from a low-level of accuracy to a high level. Initial timing closure might be done with Steiner estimates. Then global routing can be invoked and optimization will use global wires and perhaps use first order coupling analysis. Then detailed routing can be run and timing closure can proceed with accurately extracted detailed wires. As the level of accuracies increase, optimizations become more expensive to perform so most optimizations should occur with the coarsest levels of accuracy.

How to organize a “best” flow for each subsequent phase of accuracy (Steiner, global and detailed wires) to come up is still an ad hoc approach (or an art). The academic literature has not been able to address these problems because it lacks all the components and infrastructure to tune such a flow. For efficiency reasons, physical synthesis may employ heuristic approaches, starting with large changes and inexpensive analyses early in the design flow and then transitioning to more expensive analyses and restricting consideration to small changes as the design converges.

During physical synthesis, each optimization or transform generally has a special purpose, and tries to optimize one objective without hurting the others ones. A buffer insertion transform generally tries to reduce delay and electrical violations while using minimum area and power cost. A gate sizing transform can be used to fix critical paths, but it also can be used to recover area and power. One can also develop complex transforms that perform more than one optimization at a time, such as simultaneous gate sizing and Vt swapping [3], buffer insertion and layer assignment [4]. Fast incremental timing analysis is a critical ingredient for understanding whether or not transforms should be accepted. Similarly, intelligent placement services that incrementally place and legalize changes to the netlist are necessary.

While physical synthesis is a fairly mature technology, it is far from a solved problem in industry. Modern technologies (45 nm and beyond) provide a host of problems that can easily break physical synthesis. For example, complex design rules, IP/macros from hierarchical designs, the inability of buffers to drive long distances, more and varying metal layers make routability an increasingly vexing problem [5]. Physical synthesis has to be much more cognizant of not just timing closure, but of creating routable designs. Thus, new optimizations that try to spread cells, refactor logic, and find alternative buffering strategies are key to achieving routing closure.

Logic synthesis, placement, routing, and clocking are no longer truly separate tasks. The choices made by one component severely affect the others. Consequently powerful incremental techniques that can surgically attack timing and congestion problems are growing in importance. It is impossible for optimizing one objective not to sometimes mess up another, which makes recovery very important. For example, clock insertion could destroy routability, and one needs techniques to incrementally clean up damage caused by this disruption.

[1] Charles J. Alpert, Shrirang Karandikar, Zhuo Li, Gi-Joon Nam, Stephen T. Quay, Haoxing Ren, C. N. Sze, Paul G. Villarrubia, and Mehmet Yildiz, "Techniques for Fast Physical Synthesis," Proceedings of the IEEE, Vol. 95, No. 3, March 2007, pp. 573-599.

[2] Charles J. Alpert, Dinesh P. Mehta, and Sachin S. Sapatnekar, "Handbook of Algorihms for Physical Design Automation,", 2008, Auerbach Publications, Boston, MA, UA.

[3] Yifang Liu and Jiang Hu, “A New Algorithm for Simultaneous Gate Sizing and Threshold Voltage Assignment,” IEEE Trans. On CAD of Integrated Circuits and Systems, Vol. 29, No. 2, Feb 2010, pp. 223-234.

[4] Zhuo Li, Charles J. Alpert, Shiyan Hu, Tuhin Muhmud, Stephen T. Quay, and Paul G. Villarrubia, “Fast interconnect synthesis with layer assignment,” In

Proceedings of the 2008 international symposium on Physical design (ISPD '08), pp. 71-77.

[5] Charles J. Alpert, Zhuo Li, Michael D. Moffitt, Gi-Joon Nam, Jarrod A. Roy, and Gustavo Tellez, “What makes a design difficult to route,” in Proceedings of the 19th international symposium on Physical design (ISPD '10). pp. 7-12.