Keynote Address

Challenges and Principles of Physical Design

Steven L. Teig
Simplex Solutions, Inc.

INTRODUCTION

Addressing the physical design of integrated circuits, MCMs, and printed-circuit boards requires solving some of the largest and most difficult combinatorial optimization problems ever attempted. Deep submicron process technologies and the rapid and continuing increase in design size significantly add to the difficulties. Fortunately, computing power, memory, and disk also continue to increase rapidly, providing hope that our mostly sub-quadratic heuristics will keep up with problem sizes. The increasing popularity of assorted hierarchical methodologies provides additional reason for optimism.

Unfortunately, the vast majority of the fundamental techniques used in physical design today are ten to twenty years old[1]. These techniques were developed when computers had hundreds of times less CPU and storage, and they incorporate artificial and needless constraints on both designs themselves and the algorithms used to lay them out.

The physical design community includes some of the most skilled algorithmists in the world. We have an opportunity to develop new algorithms and, even more importantly, new methodologies that permit chips and systems to be smaller, faster, lower power, more reliable, and easier to design. To fulfill our potential, we must choose carefully the research directions that will have the most leverage to designers. We must also articulate principles to follow that will maximize the effectiveness of future algorithms and heuristics in our discipline. I give a personal view on these two questions below.

CHALLENGES

1) Unified routing of "special" nets and signal nets

Design a global and detailed routing strategy that handles clock, power, scan, and signal nets all together, accommodating their different requirements in the context of overall routability and electrical correctness.

2) Generalized detailed routing: topological, diagonal, gridless, no preferred direction

Design a liquid router: one that does not require Manhattan wires, a routing grid, net ordering, or an artificial preferred direction. Permit topological wiring operations. Model timing and reliability.

3) Unified floorplanning and placement

Design a global placer that can handle a mixture of a few large blocks, dozens of small blocks, and more than a million cells simultaneously. It must account for power (including multiple power levels), signal integrity, clock planning, and timing, as well as routability within various wiring models.

4) Capacity, non-SMP parallelization, and hierarchy

Design a physical design system that can support arbitrarily large netlists flat. In addition, design for parallelization over a network and support for hierarchical design methodologies to reduce turnaround times.

5) Intentional skew and intentional slew

Zero skew is an artificial constraint. Acceptable IR drop is necessary only for the parts of the circuit that are running. Design clock and power strategies that permit the design to run at speed and with the lowest power consumption, imposing as few other constraints as possible.

6) Unified synthesis, "global" placement, timing, global routing, and "detailed" placement

Combinational synthesis has not changed fundamentally in over ten years. Analytical placement, FM partitioning, and annealing as practiced today all date from the 80's, too. Almost everyone uses sequential, rip-up-and-reroute maze routing for global routing. Are we globally optimal?? Use the 500x improvement in compute power to do much better.

7) Combinatorial optimization

Almost everything that we do is ultimately combinatorial optimization. Monte Carlo and other methods continue to advance rapidly, and there is still room for revolutionary improvement. Are we going to bi-partition with FM or apply simplistic legalization to analytical placements forever?

8) Automatic metric discovery with machine learning

The complement to any optimizer is the metric to be optimized. Use machine learning techniques to discover efficient, predictive metrics automatically (or semi-automatically) instead of making them up as we do now.

9) Probabilistic tools and methodologies robust to process variations (e.g., at least 93% of these parts must run at 300 MHz)

As we venture deeper into the nanometer realm, it becomes unacceptably burdensome to over-design around putative process variations. Develop methodologies and tools that model the joint probability distribution on manufactured sub-components to advance both performance and reliability.

[1] Not to slight the many creative refinements in the last decade, but there have been very few "discontinuities".

ISPD'02, April 7-10, 2002, San Diego, California, USA.
ACM 1-58113-460-6/02/0004.

10) Asynchronous design

God does not use clock grids and trees. Why do we? Develop a physical design based on an asynchronous or semi-synchronous design methodology.

11) Cell design

Take pins off-grid. Integrate cells *and macro-cells* designed on-the-fly within P & R systems. Should components be in rows? Should we reconsider how to get power and clock to cells?

PRINCIPLES

1) Use principles!
2) Build a language in which to talk about the problem, and then solve the problem.
3) Separate the metric from the optimization technique whenever possible.
4) Make sure the metric can correctly rank any two states you can rank.
5) Great results in practical run times are usually far better than mediocre results in short run times.
6) Build adaptive systems.
7) Carefully choose assumptions about the problem domain, and exploit them.
8) Resist artificial design constraints.

SUMMARY

The design and manufacture of great electronics depends fundamentally on the physical design community: the community that made such ideas as standard cells, gate arrays, sea-of-gates, and the X Architecture *practical*. Just as the process technologists introduce new *technology nodes* every couple of years, the physical design community has the opportunity to introduce new *design nodes* more often than once per decade, as it has in the past. By exploiting the great and still rising power of computing, we can now explore approaches that would have been impossible even five years ago. We can lift artificial constraints, such as Manhattan wiring, griddedness and preferred directions; harmful distinctions, such as that between special nets and signal nets; and restrictive design methodologies, such as zero-skew clocks (or system-wide clocks at all!). We can realize a probabilistic design methodology that is robust to process variations. We can also respond to our extreme need for large-capacity combinatorial optimization techniques by advancing that field as a whole: both in the optimization strategies themselves and in a more automated search for good measures. If we do these things, we will have an even greater impact on the future of electronic design than we have already had on its past. We will facilitate the upcoming, earthshaking transition to nanotechnology by helping to make it practical, too.