Timing Closure Based on Physical Hierarchy

Jason Cong
Computer Science Department
Univ. of California, Los Angeles
Los Angeles, CA 90095 USA
cong@cs.ucla.edu

ABSTRACT

In this paper, we shall present the progress and results of the ongoing project at UCLA on synthesis and optimization under physical hierarchy. First, we shall motivate our approach by pointing out the limitations of the existing approach to interconnect planning based on early RTL floorplanning following logic hierarchy. Then, we shall discuss the technical challenges for synthesis under the physical hierarchy, including handling high computational complexity from the flattened logic hierarchy, needs of retiming and pipelining over global interconnects, and extension of existing synthesis operations. Finally, we shall outline our approaches to overcome these technical challenges.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: INTEGRATED CIRCUITS —*Design Aids.*

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Physical hierarchy, logic hierarchy, timing closure, interconnect planning, interconnect optimization, multilevel optimization, retiming and pipelining, sequential arrival time.

1. INTRODUCTION

According to the Moore's law, exponential scaling in the past thirty years has made interconnect the dominating factor in determining overall system performance and reliability. Timing closure between synthesis and layout has become a serious challenge to IC designers due to significance of interconnect delays in deep submicron technologies. To ensure timing closure it is important to consider the interconnect effect throughout the entire design process, especially in the early stages.

The prevailing thought and/or practice in today's industry is to perform early floorplanning at the RT-level (or even behavior level) on the functional modules defined in the HDL description. Then, each module is synthesized and laid out following the RT-level floorplan result and using the interconnect information defined in the RT-level floorplan. Figure 1 shows an example of the logic hierarchy and an associated floorplan, where each block in the floorplan corresponds to a functional block in the HDL description. The figure on the top shows the major

```
module cpu(pj_su, pj_boot8, …);
input …;
output …;
fpu fpu(.fpain(iu_rs2_e), .fpbin(iu_rs1_e), .fpop(fpop),
          .fpbusyn(fp_rdy_e), .fpkill(iu_kill_fpu),
          .fpout(fpu_data_e), .clk(clk), …);
pcsu pcsu(.pj_clk_out(pj_clk_out),  …);
smu smu(.iu_optop_in(iu_optop_din), …);
dtag_shell dtag_shell(.tag_in(dcu_tag_in), …);
dcram_shell dcram_shell(.data_in({dcu_din_e[31], …);
dcu dcu( .biu_data(pj_datain), …);
itag_shell itag_shell(.icu_tag_in(icu_tag_in), …);
icram_shell icram_shell(.icu_din(icu_din), …);
icu icu(.biu_data(pj_datain), …);
iu iu(.iu_data_vld(iu_data_vld), …);
endmodule
```

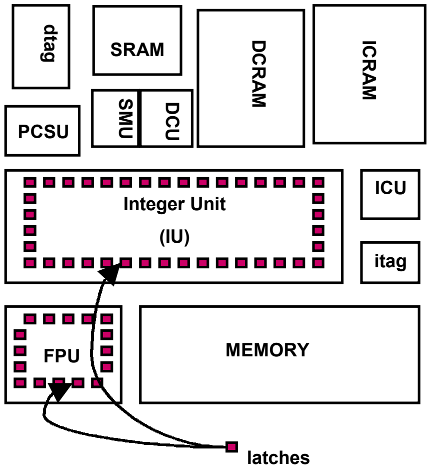


Figure 1: An example of the logic hierarchy and an associated floorplan.

ISPD '02, April 7-10, 2002, San Diego, California, USA.
Copyright 2002 ACM 1-58113-460-6/02/0004…$5.00.

functional modules in the logic hierarchy in the Verilog description of the PicoJava processor. The figure on the bottom shows one associated floorplan under this logic hierarchy.

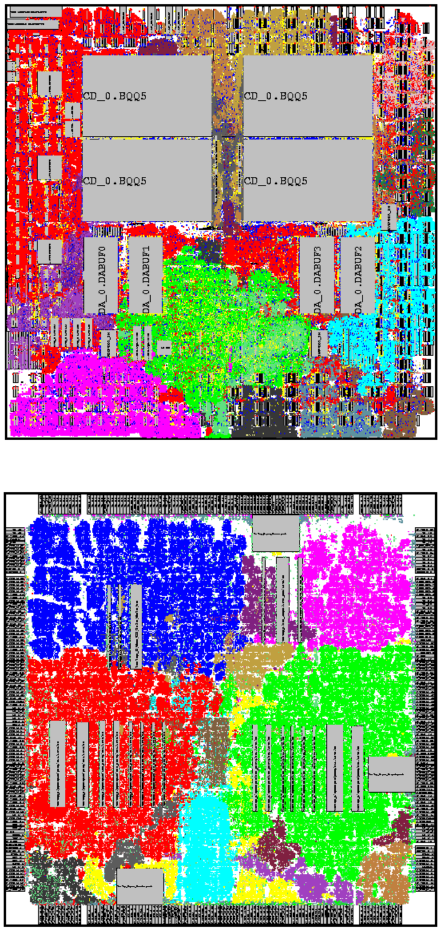
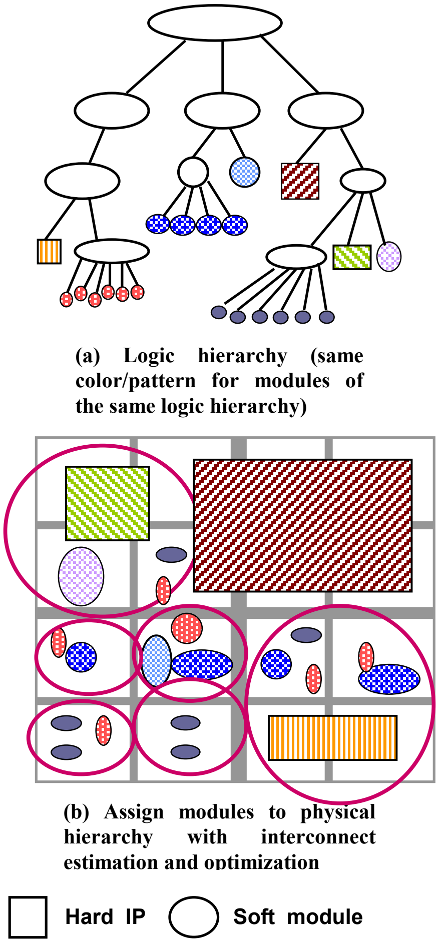


Figure 2: Two examples of logic hierarchy in the final layout (Courtesy of IBM). They are large ASIC designs with over 600,000 placeable objects, designed by using IBM's SA27E technology (a 0.18um technology with L_{eff} = 0.11 um and using copper wires).

Our study, however, raises serious doubts about the benefits of RT-level floorplanning based on the HDL description of the design, because the HDL description provided by the designer usually follows the ***logical hierarchy*** of the design which reflects the logic dependency and relationship of various functions and components in the design. Such logical hierarchy may not map well to a two-dimensional layout solution since it is usually conceived with little or no consideration of the layout information. Our concern is echoed by a number of leading experts in the industry. For example, Figure 2 shows two examples of the logic hierarchy in the final layout (obtained by optimizing directly on the flat design) of designs provided by our colleague at IBM. Modules in the same block in the logic hierarchy have the same grey shading in the layout. As can be seen, the logic hierarchy does not map directly into the physical hierarchy, as the logic with the same color does not necessarily stay together. This suggests that enforcing floorplanning to follow the logic hierarchy boundary can be harmful to the final layout.

At UCLA, we have been developing a novel approach to perform synthesis and optimization under physical hierarchy in order to achieve timing closure between synthesis and layout. In this approach, we first flatten function and logic hierarchy, so that each element in the flattened hierarchy is either a hard IP block (such a memory block) or a basic functional unit (such a 32-bit ALU) that we are certain about its physical locality. We then compute a good physical hierarchy so that it can hide as much global interconnect latency as possible. Such physical hierarchy generation in fact defines the global, semi-global, and local interconnects (based on their levels in the physical hierarchy) and has significant impact on the final design quality. Finally, we perform various


(a) Logic hierarchy (same color/pattern for modules of the same logic hierarchy)

(b) Assign modules to physical hierarchy with interconnect estimation and optimization

Figure 3: Physical hierarchy generation

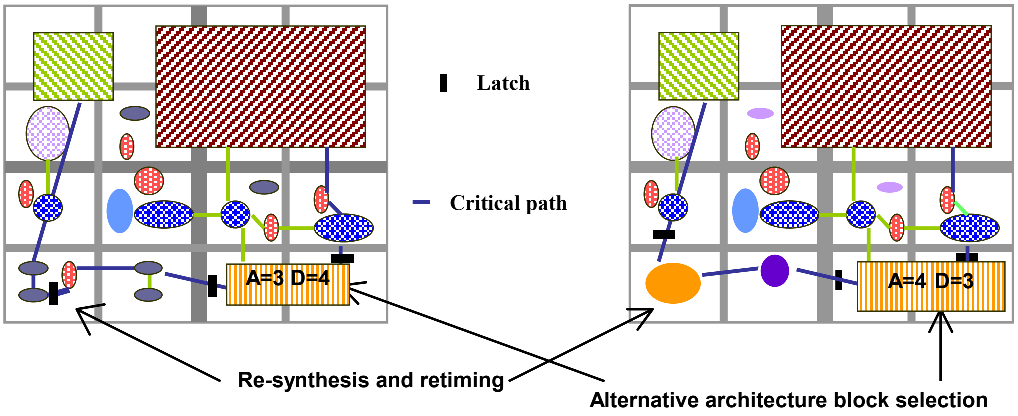

Latch
Critical path
A=3 D=4
A=4 D=3
Re-synthesis and retiming
Alternative architecture block selection

Figure 4: Synthesis under physical hierarchy

synthesis and optimization operations under the resulting physical hierarchy. Figure 3 illustrates the process of physical hierarchy generation, and Figure 4 illustrates some synthesis and optimization operations that we may perform under a given physical hierarchy.

2. TECHNICAL CHALLENGES

In order to successfully implement our approach to synthesis under the physical hierarchy, we need to overcome the following three technical challenges.

a) **Handle the high complexity of the flattened logic hierarchy:** Due to the continuous exponential scaling of devices according to Moore's Law, we expect on-chip integration of over half a billion transistors in the 0.07µm technology generation by 2008 [1]. This high design complexity makes it very challenging to handle the nearly flattened logic hierarchy to construct a good physical hierarchy.

b) **Consider retiming/pipelining over global interconnect during physical hierarchy generation:** Multiple clock cycles are needed to cross the global interconnects for multi-gigahertz designs in nanometer technologies. Figure 5 shows that 7 clock cycles are needed to go from corner-to-corner for the predicted die-size in the 0.07µm technology generation, assuming a 5 GHz clock, even with aggressive interconnect optimization including buffer insertion and wire-sizing. Therefore, we have to consider retiming and pipelining during physical hierarchy generation.

c) **Consider physical hierarchy during synthesis and optimization:** Most existing synthesis and optimization operations are developed without consideration of any physical information. We need to develop new theories and algorithms so that synthesis and optimization operations can consider the information on global and semi-global interconnects available in the physical hierarchy.

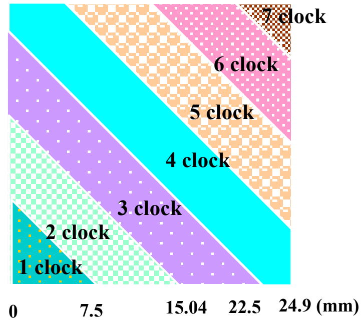


Figure 5: Illustration of how far each clock can travel to cross the global interconnects. Assume a die of 24.9mm x 24.9mm in the 0.07µm technology with a 5 GHz clock as predicted in NTRS'97 [2]. Optimal buffer insertion and wire-sizing are performed using the TRIO package [3] with the driver, buffer, and receiver sizes being 100x the minimum size inverter. Device and interconnect parameters are based on those presented in [4].

3. OUTLINE OF OUR TECHNICAL APPROACH

In this section, we shall outline our algorithms and solutions to address the technical challenges discussed in the preceding section.

3.1 Handle the high complexity of the flattened logic hierarchy

In order to handle the large problem size of the flattened logic hierarchy, we choose to use the multilevel optimization framework for physical hierarchy generation. The multilevel methods were originally used as a means of accelerating numerical algorithms for partial differential equations (e.g [5,6]). In the past decade, it has also been applied to other areas, such as image processing, combinatorial optimization, control theory, statistical mechanics, quantum electrodynamics, and linear algebra. Multilevel techniques for VLSI physical design have recently shown promising results. Good progress has been made in multilevel circuit partitioning and placement. The multilevel partitioning algorithm *hMETIS* [7] produces the best cut-size minimization in circuit partitioning. The multilevel performance-driven partitioning algorithm *HPM* [8] produces good balance of delay and cut size minimization for circuit partitioning. The multilevel placement algorithm *mPL* [9] achieves comparable circuit placement quality as the well-known *GORDIAN* package [10] with over 10x speed-up on designs with over 200K movable objects. These successes led us to choose the multilevel method for physical hierarchy generation. Another paper from our group in the proceedings describes in details our latest result on using the multilevel method for physical hierarchy generation with routability control [11]. The experimental result shows that the multilevel optimization framework leads to significant runtime improvement for physical hierarchy over the flat quadratic placement method with comparable wirelength optimization results. Moreover, it enables the use of a fast, incremental global routing engine during physical hierarchy generation for routability control, and leads to significant reduction of routing congestion.

3.2 Consider retiming/pipelining over global interconnects in physical hierarchy generation

Ideally, we would like to construct a physical hierarchy so that the communication over long global interconnects can be done in multiple clock cycles. Given this need, our goal is to construct a good physical hierarchy that can leads to the best performance *after* optimal retiming and/or pipelining over the global interconnects. There are two approaches to this problem. One is an iterative approach. In each iteration, one generates a physical hierarchy, then performs the optimal retiming and/or pipelining. After a certain number of iterations (determined by the available computation time), the physical hierarchy with the best performance after optimal retiming and/or pipelining is chosen. Obviously, this approach is not efficient, as we can explore only a limited number of physical hierarchies due to the high time complexity for physical hierarchy generation. Another approach is to find a way to directly compute a single (best) physical hierarchy that leads to the best performance after retiming and/or pipelining on global interconnects. We choose the latter approach using the theory and algorithm for *sequential timing analysis*, which was first introduced in [12], and has been successfully used in our recent work on physical planning with retiming [8,13]. Such an approach allows us to perform timing analysis and optimization during physical hierarchy generation without fixing the latch/flip-flop positions in the design. The result presented in [13] shows that considering interconnect retiming during physical hierarchy generation can hide part of the global interconnect latency and leads to over 20% delay reduction compared to the existing method on separated partitioning, floorplanning, and retiming.

3.3 Extend various synthesis and optimization operations to consider physical hierarchy information

Given a physical hierarchy, we can make a reasonably accurate estimation of interconnect delays and the critical network in the design. We need to extend various synthesis and optimization operations to make use of the knowledge on interconnect delays. For example, in the case of LUT-based FPGA mapping, once we know all the net delays or edge delays in a network, we can use the FlowMap-d [14] or EdgeMap [15] algorithms to compute a delay-optimal mapping solution. We are in the process of extending other logic optimization operations to make use of the physical hierarchy information for accurate delay estimation and performance optimization. It is quite possible that during these synthesis and optimization operations, the physical hierarchy will need to be updated to achieve better timing closure. We shall discuss some preliminary results of these operations during the presentation.

4. SUMMARY

Our study suggests that the existing approach to interconnect planning based on RT-level floorplanning on logic hierarchy may lead to suboptimal designs in terms of circuit performance optimization. We propose a new approach that performs synthesis and optimization under physical hierarchy in order to achieve timing closure between synthesis and layout. We present our research effort on addressing several important issues involved in this approach, including handling the high complexity of physical generation problems, consideration of retiming and/or pipelining over global interconnects during physical hierarchy generation, and development of placement-aware synthesis and optimization operations. This approach introduces a rich body of new and interesting research problems worthy of further investigation.

5. ACKNOWLEDGEMENTS

The students involved in this project at UCLA include Chin-Chih Chang, Ashok Jagannathan, Joey Lin, Michail Romesis, and Xin Yuan. This work is supported in part by SRC, MARCO/GSRC, an IBM Faculty Partnership Award, a grant from Intel and a grant from Fujitsu under the California MICRO program. The author would like to thank Tony Drumm from IBM, Kris Konigsfeld and

Mosur Mohan from Intel for helpful discussions on hierarchical designs. The author would also like to thank Xin Yuan for her assistance in preparing this paper.

6. REFERENCES

[1] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 1999.

[2] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 1997.

[3] Cong, J., He, L., Koh, C.-K., Pan, D. Z., and Yuan, X. Tree-Repeater-Interconnect-Optimization Package (TRIO), 1999, http://cadlab.cs.ucla.edu/~trio.

[4] Cong, J. An Interconnect-centric Design Flow for Nanometer Technologies, in Proceedings of the IEEE, vol. 89, 505-527, April 2001.

[5] Brandt, A., Multi-level Adaptive Solution to Boundary Value Problems, Mathematics of Computation, vol.31, no.138, 333-390, 1977.

[6] Briggs, W., A Multigrid Tutorial, SIAM, 1987.

[7] Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. Multilevel Hypergraph Partitioning: Applications in VLSI Domain, IEEE Trans. on Very Large Scale Integration Systems, vol. 7, 69--79, Mar 1999.

[8] Cong, J., Lim, S. K., and Wu, C. Performance Driven Multi-level and Multiway Partitioning with Retiming, in Proc. ACM/IEEE 37th Design Automation Conference, 274-279, June 2000.

[9] Chan, T., Cong, J., Kong, T., and Shinnerl, J. Multilevel Optimization for Large-Scale Circuit Placement, in Proc. IEEE International Conference on Computer Aided Design, 171--176, Nov 2000.

[10] Kleinhans, J., Sigl, G., Johannes, F., and Antreich, K. GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.10, March 1991.

[11] Chang, C.-C., Cong, J., Pan, Z. D., and Yuan, X. Physical Hierarchy Generation with Routing Congestion Control, in Proc. International Symposium on Physical Design, 2002.

[12] Pan, P., Karandikar, A. K., and Liu, C.-L. Optimal Clock Period Clustering for Sequential Circuits with Retiming, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 489-498, 1998.

[13] Cong, J., and Lim, S. K. Physical Planning with Retiming, in Proc. International Conference. on Computer Aided Design, 2-7, Nov. 2000.

[14] Cong, J., and Ding, Y. On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping, IEEE Trans. on VLSI Systems, vol. 2, no. 2, 137-148, June 1994.

[15] Yang, H. and Wong, D. F. Edge-Map: Optimal Performance Driven Technology Mapping for Iterative LUT based FPGA Designs, in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 150-155, 1994.