Physical Hierarchy Exploration of 3D Processors

Guojie Luo
Center for Energy-Efficient Computing and Applications
School of EECS, Peking University
Beijing, P.R. China
Email: gluo@pku.edu.cn

Abstract—**Most of the existing 3D designs restrict each functional module in the logical hierarchy to be on a single die, which may not generate the best 3D physical hierarchy. However, a flat 3D implementation will greatly increase the design complexity. Therefore, it is worthwhile to apply virtual 3D physical design methods for design planning at the early-design stage, instead of only performing floorplanning with existing 2D modules. In general, we are motivated to use a 3D placer to explore the benefits of removing the logical hierarchical restrictions at the early-design stage. We perform some experiments on the design planning of the LEON3 processor. Compared to a flat 3D design, planning the entire processor core on a single die brings in 10% longer wirelength, and planning the entire register file on a single die brings in 20% longer wirelength. The results help the quantitative analysis on the tradeoff between the design complexity and the cost of wirelength.**

Keywords-3D integration; 3D placement; hierarchical design

I. Introduction

Most of the existing 3D IC designs restrict each functional block in the logical hierarchy to be in a single die, which may not generate the best 3D physical hierarchy (see the discussion in [1] about physical hierarchy vs. logical hierarchy). Therefore, it is worthwhile to apply 3D-oriented physical design methods instead of only performing 3D floorplanning with existing 2D units. The study in [3] explores the 3D design space using an architectural planner and a timing and power model for 3D implementation of cache-like structures, and shows that the performance improvement and power reduction is significant when adapting 3D functional units. Thus, it is useful to consider flattening some or all levels of logical hierarchy to obtain a more optimal 3D implementation with fewer constraints on the physical hierarchy. In general, we were motivated to study the benefits of removing the logical hierarchical restrictions in 3D physical design.

In this paper, we perform physical hierarchy exploration on a netlist of the open-source LEON3 processor [5] by using a 3D placer on the netlist with different restrictions from the logical hierarchy. We can this kind of placer as *virtual 3D placer*, since it targets at the early-stage exploration instead of final implementation. Specifically, we apply the virtual 3D placer to study three kinds of designs: (i) a restriction-free design, (ii) a design with the restriction that the entire processor core is placed on a single die, and (iii) a design with the restriction that the entire register file is placed on a single die. The wirelength metric is compared among these three instances.

In the remainder of this paper, we will give a brief introduction of the placement flow of our virtual 3D placer in Section II, and then we describe the design driver we are using for this study of physical hierarchy exploration in Section III, as well as the results in Section IV. Finally, Section V concludes this paper and proposes future works.

II. Placement Flow

In this paper we focus on the analytical algorithms to solve the following 3D placement problem:

$$\begin{aligned}&\text{minimize} \quad \sum_{e\in E}\left(\mathrm{HPWL}(e)+\alpha_{\mathrm{TSV}}\cdot\mathrm{TSV}(e)\right)\\ &\text{subject to} \quad \sum_{v_i\in V}\mathrm{Area}_{m,n,k}(v_i)\le w_{\mathrm{bin}}\cdot h_{\mathrm{bin}} \qquad \text{for all } m,n,k\end{aligned}$$

where $\mathrm{HPWL}(e)$ is the half-perimeter wirelength, $\mathrm{TSV}(e)$ is the number of through-silicon vias (TSVs), and $\mathrm{Area}_{m,n,k}(v_i)$ is the area contribution of cell v_i to $\mathrm{bin}_{m,n,k}$. The 3D placement region is divided into $M\times N\times K$ bins, so that the non-overlap constraints can be formulated by the bin-wise area constraints.

A feasible intermediate placement (x_i,y_i,z_i) of cell v_i must satisfy that $x_i\in[0,W_{\mathrm{die}}]$, $y_i\in[0,H_{\mathrm{die}}]$ and $z_i\in\{1,...,K\}$. To make the analytical placement techniques applicable, the discrete variables z_i are relaxed and mapped to a continuous space. A virtual 3D placement region $[0,W_{\mathrm{die}}]\times[0,H_{\mathrm{die}}]\times[1,K]$ becomes the feasible region for the 3D global placement.

The inequality constraints in this formulation can be converted to equality constraints by adding dummy cells, which are artificial cells that only occupy area but do not connect to any other cells. Therefore, the equality-constrained optimization problem can be solved by the quadratic penalty method, which is as the following

$$\begin{aligned}&\text{step 1:}\quad \text{minimize} \quad \sum_{e\in E}\left(\mathrm{HPWL}(e)+\alpha_{\mathrm{TSV}}\cdot\mathrm{TSV}(e)\right)+\\ &\qquad\qquad \mu\cdot\sum_{m,n,k}\sum_{v_i\in V}\left(\mathrm{Area}_{m,n,k}(v_i)-w_{\mathrm{bin}}h_{\mathrm{bin}}\right)^2\\ &\text{step 2:}\quad \text{increase } \mu \text{ in a way that } \mu\leftarrow 2\mu\\ &\text{step 3:}\quad \text{repeat step 1 if } Area_{m,n,k}(v_i)>w_{bin}h_{bin}\end{aligned}$$

Please refer to [4] for the detailed implementation of a 3D analytical placement approach.

III. Design Driver

We synthesize a single-core LEON3 processor [5] with a 90nm digital cell library and a 90nm memory macro library. In Table 1, the cell number, macro number, net number and total area are extracted from the synthesized netlist, and we estimate the 2D HPWL using the Cadence SoC Encounter 6.2. To obtain the 2D HPWL, we create a square placement region with 10% whitespace, and then run the placer in Encounter without congestion effort and timing optimization for a wirelength-driven placement. In comparison, we run our mixed-size 3D placer to obtain the 3D HPWL, which is performed on a 2-die placement region with 10% total white space. The 2-die 3D implementation provides a potential of more than 40% wirelength reduction compared to the 2D implementation, with about 4000 TSVs. As an estimate of the TSV cost, we may assume each TSV consumes a 3×3 μm^2 pitch, and the capacitance of one TSV is approximately equal to an 8μm metal-2 wire [2]. With such a small amount of TSV cost, the capacity overhead is equivalent to a wirelength cost of 0.032×10^6 μm (less than 2% of the total wirelength), so the 3D implementation results in a great reduction in power.

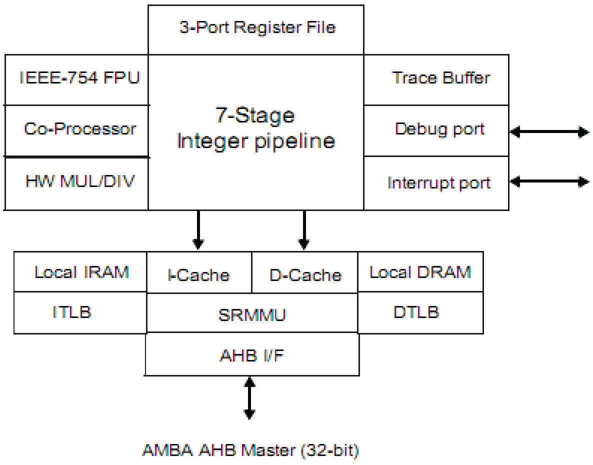


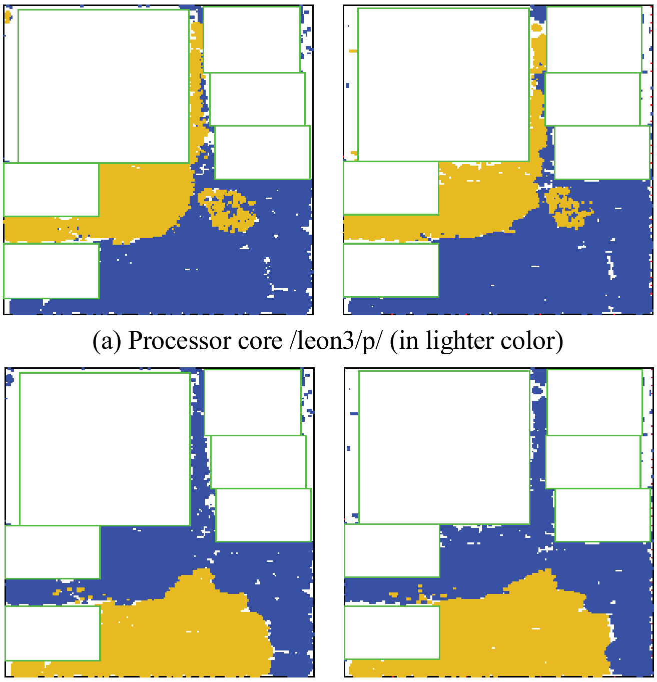
Figure 1. LEON3 processor core block diagram [5]

Table 1. Overall statistics of the synthesized netlist

#cell	#mac	#net	total area (μm^2)	2D HPWL (μm)	3D HPWL (μm)	3D TSV
34225	12	36789	6.67×10^5	1.70×10^6	0.99×10^6	3835

This 3D placement is performed with the flattened netlist, which obtains a great reduction in wirelength by removing all the logical hierarchical restrictions. The placement of the processor core and the register file is shown in Figure 2(a) and Figure 2(b), respectively. In each set of figures, the left one is the placement on the bottom die, and the right one is the placement on the upper die. The processor core and the register file are placed on both dies. It is obvious that these units are not confined to one die, and are not even in a cuboid shape.

It remains a question whether such wirelength reduction is achievable by maintaining part of the logical hierarchy. To conduct this study, we shall first summarize the logical hierarchy of the synthesized netlist. Table 2 and Figure 3 show the per-unit area consumption of the logical units. The logical hierarchy is not exactly the same as the architectural diagram in Figure 1, but there exist corresponding logical units for the blocks in the diagram. There are 12 hard macros consuming more than 60% area; these are the memory blocks for the cache memory, the TLB memory and the trace buffer. The major logical units include the processor core (/leon3/p/, 11.1%), the register file (/leon3/rf/, 16.6%), the cache memory (/leon3/cmem/, 38.1%), the TLB memory (/leon3/tbmem/, 12.0%) and the debug support unit (/dsu/, 13.4%), which are visualized in Figure 3.



(a) Processor core /leon3/p/ (in lighter color)

(b) Register file /leon3/rf/ (in lighter color)

Figure 2. Placement of logical units

Table 2. Statistics of the per-unit area consumption

unit name	area%	description
/leon3/p/	11.1%	processor core: pipeline (6.6%), MMU (2.2%), MUL (1.6%) and DIV (0.7%)
/leon3/rf/	16.6%	register file
/leon3/cmem/	38.1%	cache memory blocks: 4KB data cache (15.4%), 512B data cache tag (3.7%); 4KB instruction cache (15.4%), 512B instruction cache tag (3.7%)
/leon3/tbmem/	12.0%	TLB memory blocks (4 x 256B)
/dsu/	13.4%	debug support unit with trace buffer (4 x 256B)
/mctrl/	1.8%	memory controller
/irqctrl/	0.3%	interrupt controller
/uart/	0.7%	UART serial interface
/ahb/	2.4%	AMBA AHB bus

/apb/	1.7%	AMBA APB bus
/gptimer/	1.4%	general purpose timer unit
/grgpio/	0.3%	general purpose I/O port

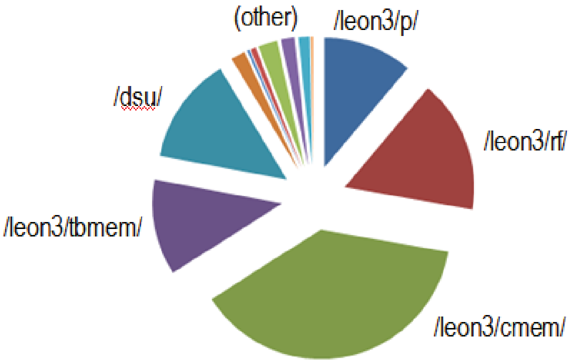
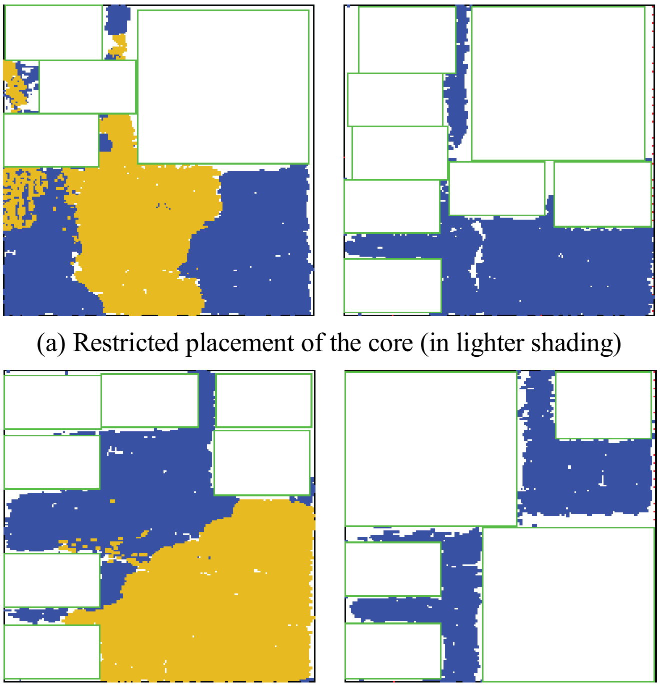


Figure 3. Plot of the per-unit area consumption



(a) Restricted placement of the core (in lighter shading)

(b) Restricted placement of the register file (in lighter shading)

Figure 4. Restricted placement of some logical modules

IV. Physical Hierarchy Exploration

To restrict one logical module placed on only one die, we first create addition variable constraints in the problem formulation in Section II, such that the cells in the restricted module must with the same value. We than run the mixed-size 3D placer to obtain the placement with restrictions. Figure 4 shows two results: Figure 4(a) is the 3D placement which is a result of placing the entire processor core on a single die, and Figure 4(b) is the 3D placement which is a result of placing the entire register file on a single die. The HPWL and TSV number are compared in Table 3, which shows that the restricted placement for the processor core brings in 10% longer HPWL, and the restricted placement for the register file brings in 20% longer HPWL.

Table 3. Placement results with different restrictions

Figure 2		Figure 4(a)		Figure 4(b)	
HPWL	TSV	HPWL	TSV	HPWL	TSV
0.99×10^6	3835	1.09×10^6	1715	1.20×10^6	845

V. Conclusions and Future Work

In this paper we perform a case study of the physical hierarchy exploration of the 3D designs of the LEON3 processor. Compared to a flat 3D design, planning the entire processor core on a single die brings in 10% longer wirelength, and planning the entire register file on a single die brings in 20% longer wirelength. This kind of data will help chip architects to design whether it is worthy to plan one logical module on a single die.

In the future, we are going to automate the physical hierarchy exploration process for multi-core designs, where additional regularity constraints will be imposed. The regularity constraints require that the placement of the same type of modules is unique, so that these modules with multiple occurrences can be implemented and tested only once to reduce the design complexity. Quantitative analysis will be made on the tradeoff between the design complexity and the cost of placement quality.

References

[1] J. Cong, "Timing closure based on physical hierarchy," *Proceedings of the 2002 international symposium on Physical design*, p. 170, 2002.

[2] W. R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical," *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 498-510, Jun. 2005.

[3] Y. Liu, Y. Ma, E. Kursun, G. Reinman, and J. Cong, "Fine grain 3D integration for microarchitecture design through cube packing exploration," *Proceedings of the 25th International Conference on Computer Design*, pp. 259-266, Oct. 2007.

[4] G. Luo, "Placement and design planning for 3D integrated circuits," *Ph.D. dissertation*, University of California, Los Angeles, 2011.

[5] http://www.gaisler.com/