Issues in the Floor Planning and Layout of Asynchronous VLSI Systems

3N−1

Metehan ÖZCAN Masashi IMAI Hiroshi NAKAMURA Takashi NANYA

Research Center for Advanced Science and Technology, The University of Tokyo

1 Introduction

Scaling integrated circuit (IC) dimensions has a great impact on VLSI systems. Although switching speed can be improved significantly, interconnection delays are, and will be more in future, the major reason for performance degradation. Nowadays, especially in deep submicron design, interconnections are responsible for most of the signal delay, even up to 90 percent [1].

In last decades speed-ups in performance of processors were due to changes or improvements in computer architecture and advances in device technologies. However, in recent years new processors gain much of their speed mainly owing to increases in clock frequency. Instruction level parallelism is limited due to memory latency, dependencies, and difficulties in branch prediction. Therefore, it is expected that in future new processors will exploit performance more and more from clock frequency (or in other words frequency of operation) rather than microarchitecture features.

Due to its average-case behavior and nonexistence of clock skew, asynchronous systems are suggested as an alternative for current synchronous systems. Semiconductor Industry Association suggests that in 2003 at 0.13μm technology asynchronous architecture will be prominent [2].

2 Layout process

The general steps in the design of a VLSI is shown on the left-hand side in Figure 1. Layout corresponds to the last three steps, i.e. floor planning, placing, and routing. This figure shows only the general steps, actually verification is needed almost after all the steps. Also there may be interactions and feedback between steps. In addition, earlier and more accurate estimation of wires delays are needed, this is specially important for global interconnections.

Floorplanning has two main steps, i.e. partitioning the design and placing the blocks or clusters. Partitioning can be done in two ways, either by clustering, a bottom-up approach, or using a top-down partition method, such as min-cut partitioning. Since the current applications are too big in size, a hybrid approach of bottom-up and top-down methods is more promising. First bottom-up clustering may be applied to reduce the number of elements to be dealed by partition algorithms.

Although after the design there exist a logical hierarchy and grouping, this logical hierarchy is to be converted and transfered to physical hierarchy. And since there are much less levels in physical hierarchy compared to logical one, this process is usually known as flattening the design.

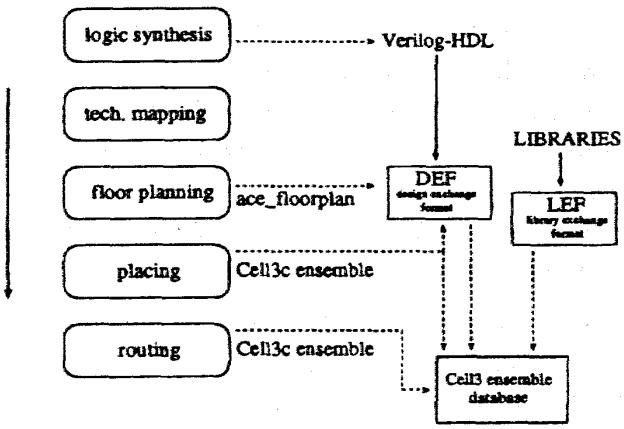

Figure 1: TITAC-2 [4] design flow.

In the layout society, there is a special trend for interconnection layout, i.e. "design for interconnectivity," which is based on the prediction of likely interconnection routes and incremental approach for design and layout [3]. However, applying this methodology is not straightforward, especially in scaled technologies; because the distributed nature of the interconnections and cross-talk become important in modeling interconnections. Therefore, more accurate *RC* tree, *RLC* tree, and Elmore delay models are used.

Furthermore, driver sizing, transistor sizing, topology optimizations, wiresizing optimizations and simultaneous applications of these methods have been used in recent works for interconnection layout. These methods can be very effective specially for routing.

3 Layout for asynchronous design

Although asynchronous architecture and logic design

should be researched in detail, it is also important to enhance the layout according to the needs of asynchronous design methodologies, especially in scaled technologies.

Nonexistence of clock lines, excess amount of C elements, request and acknowledgment signals, and average case behavior are the main differences of asynchronous circuits from synchronous counterparts.

When the new models or new methodologies are included in the layout tools they can also be used in asynchronous design methodology. For example, currently Steiner trees are being used by Cadence's Cell3 ensemble tool to model interconnections. If that model is improved it can be directly used by asynchronous design too.

TITAC-2 is a 32-bit asynchronous microprocessor in 0.5μ rule. Figure 1 shows the tools used for the design of TITAC-2 [4]. Logic synthesis done in Verilog format is first transferred to Design Exchange Format. Also standard cell libraries and macro cells are transferred into the Library Exchange Format. After the floorplanning is done by ace_floorplan, Cell3 ensemble engine is used for placing and routing.

In the design of TITAC-2 hierarchical place and flat route methodology is used for layout.

4 Suggested methodology and challenges

Since developing a layout tool from the scratch is not feasible at the beginning, a nice approach is first adjusting the floor planning according to the asynchronous design methodologies. After proper updates to floor planning are fully understood, necessary alteration in the lower levels of design flow in Figure 1 can be performed in the next steps of research. These updates may later be automated to be able to deal large scale circuits.

In asynchronous design, signals,especially request and acknowledgment signals, are collected by C elements. An interesting idea could be developing algorithms which are to do floor planning or partitioning by specially considering C elements, and later placing and routing C elements and request and acknowledgment lines specifically. Furthermore, while partitioning the circuit, average case behavior of the asynchronous circuits can be taken into account. Different weights can be given to different parts accordingly.

There are two alternative ways to do these updates; one is to develop a stand-alone floor planning algorithm, the second one is using an existing floor planner and making the necessary updates on that tool. In our case; the Cadence's Preview floorplanner and automatic analysis environment engine is being used for the latter approach. Necessary and appropriate updates peculiar to asynchronous design is being researched. Also partitioning and timing constraints can be imposed accordingly.

A big challenge for asynchronous layout is the handling of global interconnections. Actually the same challenge also exist for the synchronous counterpart. However, that is more severe for the asynchronous one. In global level request and acknowledgment scheme is usually used for the communication of high-level modules. But this means that a two-way communication is occurring, which is very costly in global level. This issue further increases the importance of floor planning and layout for the asynchronous design. Floor planning maybe performed around the request and acknowledgment lines, and they are to be shortened as much as possible. In this context, new floorplanning algorithms or using existing suitable ones, like spiral floorplans [5], may be necessary. Even this problem may lead a very novice and yet undiscovered delay model in global level. Note that in the low-level this overhead may not be that much significant, where modular design, reusability, correctness, and average behavior may compensate that overhead. In addition, global interconnections become more interesting and challenging in case of pipelining, especially when deep pipelining is concerned.

After developing the asynchronous floorplanner, next steps could be developing new asynchronous place and route algorithms, or say update algorithms, according to the above methodology.

Bibliography

[1] T. C. Lee and J. Cong, "The new line in IC Design," *IEEE Spectrum*, pp.52-58, March 1997.

[2] "The national technology roadmap for semiconductors," **http://notes.sematech.org/roadmap5.pdf**, *Semiconductor Industry Association*, 1997.

[3] J. Cong, L. He, C. K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integration, the VLSI Journal*, Vol.21, pp.1-94, 1996.

[4] T. Nanya et al., "TITAC-2: A 32-bit scalable-delay-insensitive microprocessor," *HOT Chips IX*, Stanford, pp.19-32, Aug. 1997.

[5] T. C. Wang and D. F. Wong, "Optimal floorplan area optimization," *IEEE Trans. Computer-aided Design*, Vol.11, No.8, pp.992-1002, August 1992.