Hierarchy -- A CHDStd Tool for the Coming Deep Submicron Complex Design Crisis

S. Grout, G. Ledenbach, R. G. Bushroe, P. Fisher
SEMATECH, 2706 Montopolis Dr, Austin, TX 78741, USA

D. Cottrell, D. Mallis
Silicon Integration Initiative, 4030 W. Braker Ln, Suite 550 Austin, TX, 78759, USA

S. DasGupta
IBM Corporation, 11400 Burnett Rd Austin, TX 78758, USA

J. Morrell, J. Sayah, R. Gupta, PT Patel, P. Adams
IBM Corporation, B/334-2, East Fishkill, New York, USA

Abstract -- This paper describes the use of a hierarchical design representation standard, CHDStd, as part of the architecture of the Chip Hierarchical Design System (CHDS). Details are given on CHDStd-based hierarchy mechanisms and processes required to support Forward Timing-Driven Hierarchical Design capabilities needed for chip design using 0.25u - 0.18u technologies and beyond. These capabilities solve some of the key challenges identified by the semiconductor industry's Design Productivity Crisis. This paper identifies the role of hierarchy for handling difficult chip design information issues and for large complex chip design.

I. Introduction

SEMATECH, a consortium focusing on advanced semiconductor technology, is developing a new design system, Chip Hierarchical Design System (CHDS) for design teams to design large complex chips for 0.25-micron technologies and beyond. The need for improved EDA capability can be traced the Design Productivity Crisis (DPC) [1] identified in the 1994 National Technology Roadmap for Semiconductors (NTRS) [2]. Growing complexity in chip design is the result of shrinking feature size, increasing functionality, and complexity of large complex hierarchical designs. Figure 1 illustrates the DPC with chip complexity growing 58% annually. Design productivity currently is growing 21% annually but predicted to slow with future technologies.

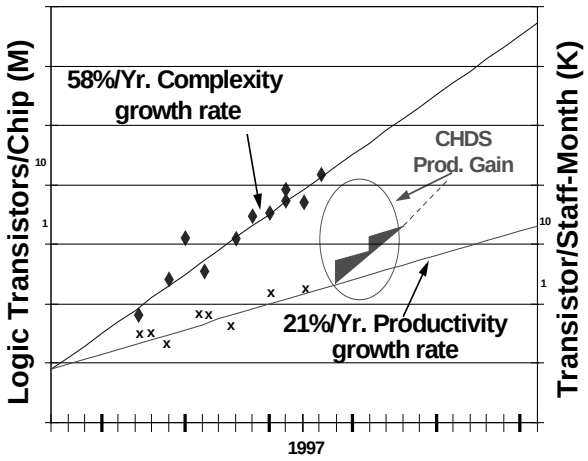

Fig. 1 - Design Productivity Crisis

Future designs will need significant increases in productivity in order to support greatly increased chip complexity and design efforts. One of the key strategies to keep growth under control is use of greatly improved design representation and, in particular, use of a hierarchy within the design representation through development of a proposed high performance hierarchical chip design standard, CHDS Technical Data (CHDStd). This paper focuses on specific needs, use and benefits of hierarchy facilities within CHDStd for designing upcoming large complex deep submicron chips.

Section II describes the problems faced today for complex design systems. Section III describes basic CHDS requirements for CHDStd hierarchical capability. Section IV describes various requirements for design hierarchy. Section V describes hierarchy divergence information handling within CHDStd. Section VI summarizes results of applying CHDStd heirarchy technology within CHDS. Section VII summarizes conclusions about CHDStd advances using hierarchy.

II. Problems Faced By Today's Chip Design Systems

Overall, today's design systems do not possess a strong ability to support large complex chip designs, designed by large design teams, with rapid timing-driven design iterations. These systems do not handle design hierarchy decomposition seamlessly, nor do they provide the ability to drive accurate physical parasitic timing back up to higher levels in the design process (e.g. architectural, behavioral or RTL design levels) so that physical effects can be first predicted and then reverified higher in the design hierarchy and with earlier design stages. This can only be achieved if knowledge of the evolution of the design interconnection is available down through the design hierarchy, as detail grows, across all stages of design.

This paper details CHDStd hierarchical functionality needed to solve design and EDA problems in the areas of logical and physical design hierarchy divergence and management, and design hierarchy tracking backannotation.

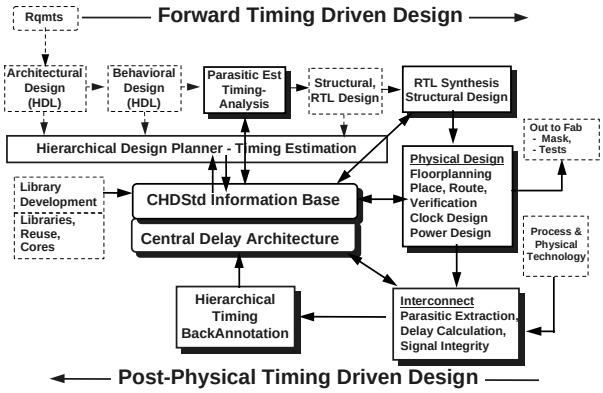

Fig. 2 - CHDS Overall architecture

III. CHDS ARCHITECTURE FOR 0.25U CHIP DESIGN

Fig. 2 shows the overall level architecture of CHDS. At first glance, it seems to be classic top-down approach common to many existing systems. However, there are fundamental differences between the details of CHDS and the EDA support for complex chip design we have today. The problem that CHDS system for timing driven logical and physical design is being challenged to solve is that chip designs with up to 28M transistors must be done in the same period of time with approximately the same design team size as for today's smaller chips, and still meet the tighter timing constraints that are required by such designs. This implies at least a 4X increase in designer productivity per process generation.

To achieve this goal, CHDS and CHDStd must provide capability that works seamlessly on any design abstraction level, including RTL, behavioral and above, be able to apply physical, timing, and power constraints early while planning the design, and drive them forward in the design process. For instance, early hierarchical physical design planning [3,4] within CHDS allows the design to be decomposed and partitioned recursively during early high-level design stages, with related physical, timing and other performance constraints apportioned accordingly.

The design must be hierarchically defined throughout the entire design process so that designers can focus on a smaller number of blocks or subdesigns at any given time and at any level of the design hierarchy. This will often also allow the tools to run faster. It is essential that hierarchy be defined and properly implemented. Given the expected complexity of designs, there often cannot be just one simple description of a design hierarchy; more likely there will be several. For instance, there may be a dominantly-logical hierarchy that a chip architect uses to view a design from the system point of view. A chip designer may prefer to focus on a dominantly-physical hierarchy that shows the partitions and critical interconnections of the chip. CHDStd must support this mix of potentially diverging logical and physical design hierarchies. CHDStd must manage correlation of design information between design hierarchies.

IV. GENERAL REQUIREMENTS FOR DESIGN HIERARCHY

At any point in the design process, both design teams and EDA tools can only deal with one contiguous portion of the design hierarchy. That is, there must be a continuity of the hierarchical netlist and the rest of the associated design data for any EDA tool to be able to work with the data. This is fundamental as logical, behavioral, and physical design data are all integrated together with a contiguous hierarchical netlist. Fig. 3 illustrates a typical chip design hierarchy.

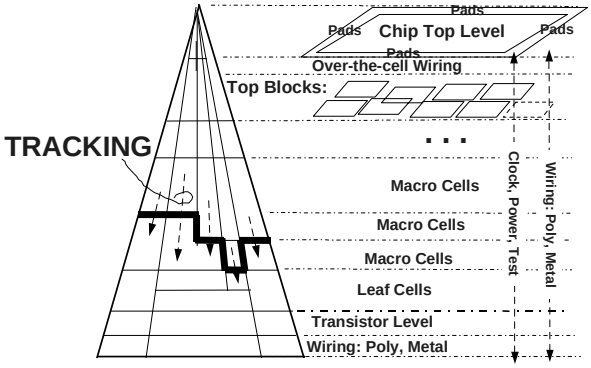

Fig. 3 - Typical Chip Design Hierarchy with 'Tracking'

To be clear in dealing with this subject, we first define logical and physical design hierarchy: A *logical design hierarchy* includes, among other logical front-end design information, a hierarchical netlist describing the block decomposition of a design ('things made of things'), with possible top-level block estimated placement and timing. This information is, of course, developed during the early part of the design process. A *physical design hierarchy* consists of a block hierarchy and associated interconnect including all detailed physical design information such as floorplan, block placement, and global and detailed routing. Note that it still has 'logical' hierarchical netlist details and is concerns the same chip design.

A. Relating Contiguous Logical and Physical Design hierarchies

Taking the above into account, to establish connectivity (join) between any part of one (e.g., logical or mixed) design hierarchy and another (e.g., physical), the hierarchical connectivity may only be stated in the following ways: 1) through wiring the two hierarchies together at higher design levels, 2) defining them as contained within the same CHDStd configuration still with explicit connectivity, or 3) defining the actual "traceability" (discussed later) between one design hierarchy and another. This is true regardless of the design information to be connected or otherwise related between design hierarchies.

B. Connectivity At Tapeout

Tapeout in CHDS supports use of the traditional GDSII format which, of course, includes no explicit connectivity, much less hierarchical connectivity. Since CHDStd includes all detailed physical routing and hierarchical netlist information., backend CHDS tools (e.g., DRC, HTB) therefore work from CHDStd, and not from GDSII.

C. EDA Tool Needs for Locally Contiguous Hierarchy

Typically, a design team will intend to create one overall design hierarchy that runs from the top of their design hierarchy to the bottom. The overall design hierarchy, for various reasons (methodology, tool limitation, designer's own activities), often has discontinuities in it. This is not necessarily disastrous as the real requirement is that each EDA tool be able to find the portion of the design hierarchy appropriate for the processing that the tool does. For instance, a detailed router will be able to work just fine if the contiguous portion of the design hierarchy it works from contains everything the router needs. Therefore, the design information must be created so that the design hierarchy is LOCALLY contiguous for the needs of each particular tool.

D. CHDStd Support for Large Teams Working Concurrently

When there are many designers working concurrently at various points in the design hierarchy, and at various stages of completion, the overall design hierarchy must at all times be globally contiguous. Prior to CHDStd, when this is not the case, multiple design teams of a large design group had to follow today's methodology of continuously manipulating design files in order to have useful data across each team's design activities. CHDStd provides means to avoid this problem area with large-chip large design teams.

E. CHDS Support For A Single Logical / Physical Design Hierarchy

There have always been designs and design processes that used a single design version and design heirarchy at any one time. In those cases, the physical information is an integrated part of the logical design information. An example of this situation is when a design process flows a design down from architectural to subsystem to behavioral to RTL design, RTL synthesis occurs, then floorplanning, placement, and detailed route, all without a break in the design decomposition. These types of design processes often may occur in small ASIC design processes, as well as occasionally in extremely large chip (e.g., microprocessor) designs. This is often the characteristic of a design process that doesn't want to (or cannot) tolerate the time and resources needed to manage design hierarchy divergence.

F. Separate Logical and Physical Design Hierarchy

As already discussed, there have always been designs and design processes that created and/or utilized more than one design hierarchy. This situation has occurred typically for two reasons:

Design Limitations: The design hierarchy has a break typically from the physical design being developed without keeping the logical design uptodate at the same design level. The detailed implementation of a design may very well have a very effective solution through use of some other netlist architecture. CHDS and CHDStd are required to support such flows regardless of the type of information involved, without extra effort on the part of the designers or main CHDS/EDA tools, and without constraining the design methodology.

EDA Limitations: In the past, EDA tools often were only capable of describing the logical and physical design in separate design files types. However, it becomes increasingly difficult to keep logical-design files in sync with physical-design files. Also, late in the design, the focus is often almost entirely on the physical design with few resources made available for keeping logical design files uptodate at the physical design level. Again, CHDS and CHDStd should not impose limitations on the design methodology, on what designers do, or on what the EDA tools can do.

G. CHDS Requirements For A MIXED Logical and Physical Design Hierarchy

For CHDStd to provide consolidated and consistent support for the above range of needs, CHDStd needs to provides means to have a mixed integrated logical and physical design hierarchy. Since physical-design information must contain an imbedded hierarchical netlist, by providing a mixed integrated logical and physical design representation, the imbedded netlist may be the same netlist as required for the logical design. CHDStd Requirements document [5,6] shows a high level information model of a mixed integrated design representation. Other CHDS design representation needs lead to CHDStd [7,8]needing to provide support, i.e., early high level physical design planning and timing estimation done concurrently with front-end logical architectural and behavioral design. In all these cases, there must be one coordinated netlist that applies to all the design data and constraints at that level.

As the design flows down to more detailed design levels (e.g., behavioral and/or RTL), the physical design and timing design needs to be kept consistent with, and uptodate with the logical design. Once the design has been synthesized down to the cell and macro cell level, detailed physical design (e.g., floorplanning, placement, routing, clock and power design) begins. To provide for static, dynamic, and other verification means at one level of design detail, a consistent overall mixed integrated design needs to be maintained. For the least amount of designer, tool, and EDA system effort, data population, traceability and ECO complexity, it is often better to deal with one design hierarchy at any one step of the overall design process.

H. Design Hierarchy Divergence

Occasionally during a real design process, there may be a discontinuity in the hierarchical netlist. An example of this might be when a designer decides to implement a portion of the design using some other computer architecture and/or

mathematical paradigm, e.g., implementation of some "if-then-else" portion of the control logic design as a state machine. Such discontinuities are, therefore, often natural. Design steps often grow out of design process discontinuities, though, that lead to the design hierarchy becoming disjoint and, more importantly, staying disjoint. The EDA tools and/or designers may continue to refine the design at more than one level or in multiple hierarchies (e.g., detailed routing and editing). Also additional physical design detail may be added without regard to the higher-level netlist or hierarchical design level being worked on.

Design hierarchy divergence is not, therefore, about two really different design hierarchies, but about difficulty of the design system and designers simply keeping all the details of design changes uptodate, regardless of the design processes. With proper capabilities, divergence may be either avoided or more easily handled than now.

V. CHDStd Support For Navigating Across Hierarchy Discontinuities - 'Tracking'

We need, then, to navigate across any discontinuities in the mixed design hierarchy. To handle that and to achieve the design effectiveness built into CHDS and CHDStd, the following CHDStd capability is provided. The need for continuity across the design hierarchy leads to the CHDS requirement to identify and record the path or TRACEABILITY across any discontinuity in the design hierarchy. As shown in Fig 3., 'Tracking' (or allocation) information is captured by a tool (or manually from the designer) and identifies how some (e.g., lower level) design aspects implement an aspect of the current block definition, e.g., how a port is mapped in some non-obvious way in the block-instance connection(s) in some lower level or separate design hierarchy. Note that when hierarchical decomposition is straight forward, the hierarchical instantiation information is entirely sufficient so that 'tracking' data need not be recorded.

By 'Tracking' these "correspondence points" as timing constraints and assertions are migrated down in the hierarchy, we can later backannotate accurate design information back up the design hierarchies and do accurate design reverification or resynthesis.

VI. Results

The proposed CHDStd hierarchical design representation technology reported here has been implemented by the authors and other EDA development teams in several design systems, and previously used for the design of large complex electronics products and chips. The 'MUSER' hierarchical design system was implemented for large computer systems design within Honeywell (now Bull). EDA efforts within MCC using these techniques were capable of reducing 1M to 10M gate designs down to some 60,000 hierarchical instances or fewer, assuming approximately 500 internal block designs (CHDStd block_definitions) and 500 leaf cells and device definitions.

SEMATECH's base technology for CHDStd has been in ongoing use within IBM and several other companies and successfully used for leading-technology large complex chips. CHDStd is at present being used to implement the CHDS system, and will be made available for general use by the SEMATECH consortium member companies and, once commercialized, by industry at large.

VII. Summary and Conclusions

The proposed CHDStd standard will be able to accommodate a range of design methodologies needed to implement upcoming target 0.25u and beyond designs. CHDS and CHDStd supports separate logical and physical design hierarchy as well as support, whenever needed, a integrated logical and physical design hierarchy. This is achieved by providing means for having a mixed logical and physical hierarchical design representation and providing traceability to precisely identify how to navigate across hierarchical discontinuities.

CHDStd hierarchical design capability for CHDS therefore comprises both a near-term solution and a long-range vision for meeting the Design Productivity Crisis for 0.25u technologies and beyond. The detailed CHDStd capabilities include significant enhancements needed for hierarchical logical and physical design, parasitic extraction, timing calculation and signal integrity checking to ensure that designs can be implemented to meet functional and performance requirements at every level of design. Significantly more flexible design processes are enabled as EDA tools implement and use CHDStd hierarchy capability to their advantage.

VIII. References

[1] "Design Needs for the 21st Century: White Paper," Semiconductor Research Center (SRC), September, 1994

[2] The National Technology Roadmap for Semiconductors, pp 5-43. Semiconductor Industry Association (SIA), 1994

[3] J. Sayah, R.Gupta, D. Sherlekar, P. Honsinger, J. Apte, S. Bollinger, H. Chen, S. DasGupta, E. Hsieh, A. Huber, E. Hughes, Z. Kurzum, V. Rao, T. Tabtieng, V. Valijan, D. Yang. Design Planning for high-performance ASICs, *IBM Journal of Research and Development*, Vol. 40 No. 4, pp. 431-452, July, 1996

[4] R.G. Bushroe, S. DasGupta, A. Dengi, P. Fisher, S. Grout, G. Ledenbach, Nagaraj NS, R. Steele. Chip Hierarchical Design System (CHDS): A foundation for Timing Driven Physical Design into the 21st Century, to be published in the *International Symposium on Physical Design* (ISPD97), April 13-15, 1997

[5] Requirements Document - SEMATECH ECAD Program CHDS Technical Data (CHDStd) - Requirements for a Chip Hierarchical Product Design Specification Needed to Support 0.25-micron Chip Design. SEMATECH, October 6, 1995

[6] Chip Hierarchical Design System - Integrated Hierarchical Forward Timing Driven Electrical and Physical Design Requirements for 0.25-micron Chip Design, Version 1.0. SEMATECH, February 21, 1996

[7] An Integrated Data Model for Hierarchical Design Assembly, May 31, 1996, unpublished - www.cfi.org/CHDStd/infomodel

[8] CHDStd Information Model Documentation, April 10, 1996, unpublished, www.cfi.org/CHDStd/chdsIndex.html