



The Design Constraints Description Language

AN EMERGING OVI STANDARD WITH WIDESPREAD INDUSTRY SUPPORT

Open Verilog International is developing the Design Constraints Description Language (DCDL), a new standard for initial entry and inter-tool exchange of constraints. At DAC '99, we're pleased to demonstrate significant support from a number of EDA tool companies for a preliminary portion of DCDL that addresses the timing domain.

Participants in the demonstration include Cadence Design Systems, Exemplar, IBM, Mentor Graphics, and Xilinx. Additional organizations contributing to the definition of the standard include Averstar, Fujitsu, Infineon, LSI Logic, Sematech, Synopsys, Texas Instruments, and the Virtual Socket Interface Alliance.

The focus of the demonstration is on illustrating the use models for DCDL, how it will improve interoperability in design flows, and the style of the language itself.

WHAT IS DCDL?

The DCDL standard defines a syntax for describing constraints and comprehensive semantics for interpreting them. The style of the syntax has been chosen to support embedding DCDL constraints in TCL scripts and in the System Level Design Language.

DCDL will address a variety of different constraint domains, including timing, clocking, area, power, test, environment/operating conditions, logic architecture, signal integrity, physical implementation, cost, reliability, and manufacturability.

WHY IS DCDL IMPORTANT?

DCDL is expected to be a cornerstone of next-generation flows. In very deep sub-micron designs, it will be crucial to be able to describe all of the different constraints on a design, optimize the design to satisfy multiple conflicting objectives, then verify that the constraints have been met.

DCDL will also be essential in IP packaging for system chip design, where it will be used as a vendor-neutral format to capture internal constraints on soft and firm virtual components.

WHO WILL USE DCDL?

Designers will use DCDL as a single, consistent basis for describing their design intent. EDA tool developers will use DCDL as a standard base for reading, writing, and interpreting constraints.

IP providers will use DCDL to describe their intent for partially implemented IP blocks, and to describe restrictions on how IP blocks may be used. IP integrators will use DCDL to complete the implementation of IP blocks. Semiconductor vendors will use DCDL as part of tool qualification, and in design kits.

WHEN WILL DCDL BE AVAILABLE?

The DCDL draft standard for the timing domain will be complete in the second half of 1999 and should go through balloting in early 2000. In 2000, the Design Constraints Working Group will add coverage for the power and signal integrity domains.

USE MODELS

There are many ways in which DCDL will be used. Two of the most important are

- Users will write application-specific command scripts, embedding DCDL statements that describe the constraints on the design
- Tools will generate constraint files using pure DCDL to describe the constraints in an application-independent form

These use models are illustrated in the two flows we're demonstrating at DAC.



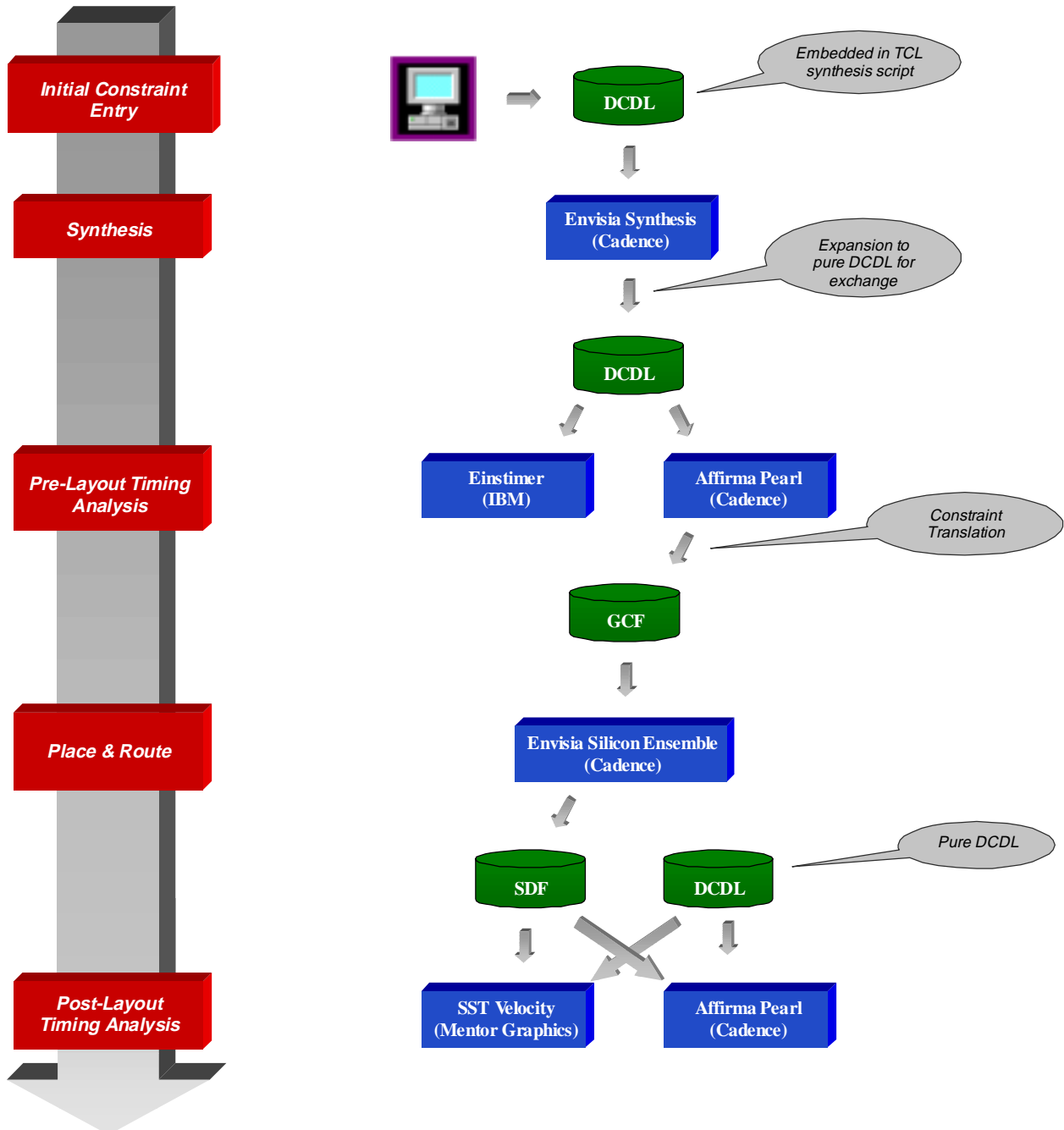
THE ASIC DESIGN FLOW

The ASIC Design Flow demonstration shows how timing constraints entered by a designer in a TCL synthesis script can be used to drive all of the timing-driven implementation tools through the rest of the flow.

The ASIC Design Flow is based on tools from Cadence Design Systems, Inc. for synthesis, timing analysis, and place & route, and on tools from IBM and Mentor Graphics for timing analysis.

The initial constraint entry is done as embedded DCDL commands within a synthesis control script. These embedded constraints are interpreted, then written out in a pure DCDL file that is read by each of the timing analysis tools.

For the demonstration, the DCDL constraints are converted into tool-specific commands by a TCL emulation layer. For place and route, the constraints are also converted into Cadence's General Constraint Format (GCF). In production use of the standard, we expect that DCDL will be supported directly by many tools.



THE FPGA DESIGN FLOW

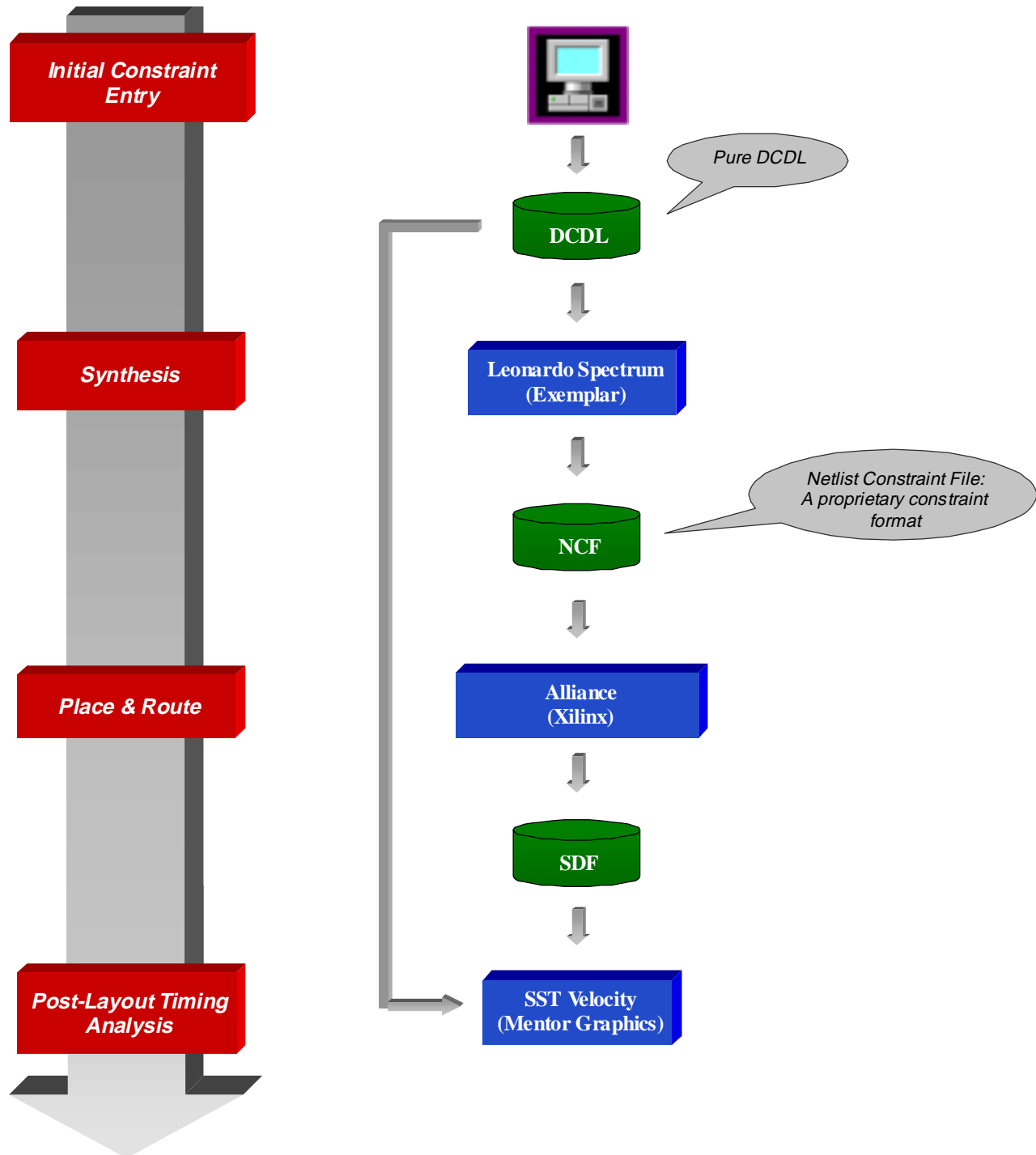
The FPGA Design Flow demonstration shows how timing constraints entered by a designer in DCDL can be used to drive implementation and timing verification of the FPGA.

The FPGA Design Flow is based on tools from Exemplar for synthesis, Xilinx for place & route, and Mentor Graphics for timing analysis.

In the FPGA flow, initial constraint entry is done with pure DCDL, which is converted into tool-specific commands by a TCL emulation layer.

For place and route, the constraints are also converted into a proprietary constraint format.

In production use of DCDL, we expect that the DCDL constraints will be supported directly by many tools.



SUMMARY

In both the ASIC and FPGA demonstrations, DCDL provided constraint interoperability across an entire design flow from RTL through post-route analysis.

The ASIC design flow included 5 different tools from 3 vendors, while the FPGA design flow included 3 different tools from 3 vendors. The demonstration clearly shows a strong industry commitment to support the standard when it becomes available in late 1999.

MORE INFORMATION

The OVI Design Constraints Working Group (DC-WG) will be holding a “Birds Of A Feather” meeting at DAC on Wednesday, June 23rd from 6:00 to 7:30pm at the New Orleans Hilton. To attend, please sign up on the board at the Information Desk.

The working group meets weekly by teleconference. To attend or to get more background on DCDL and the DC-WG effort, please see our web site at <http://eda.org/dcwg>.

Cadence, Pearl, and the Cadence logo are registered trademarks, and Envisia, Affirma, and Silicon Ensemble are trademarks of Cadence Design Systems, Inc.

Exemplar Logic, its Logo, and LeonardoSpectrum are trademarks of Exemplar Logic Inc., a wholly owned subsidiary of Mentor Graphics Corporation.

IBM and the IBM logo are registered trademarks of the IBM Corporation

Mentor Graphics is a registered trademark, and SST Velocity is a trademark of Mentor Graphics Corporation

Xilinx and the Xilinx logo are a registered trademarks, and Alliance is a trademark of Xilinx, Inc.