Design Constraints Working Group

Mark Hahn, Chair

Cadence Design Systems, Inc

mhahn@cadence.com

(408) 428-5399
(408) 428-5959 (Fax)

Agenda

- Introduction
- History
- Organizational Structure
- Charter
- Plans
- Current Status

Why Develop A Constraint Standard?

- Tower of Babel today
  - Many different formats for describing constraints
  - Inconsistent syntax
    - Requires re-entering or translating constraints
  - Inconsistent semantics
    - May not be able to translate constraints
    - Contributes to lack of convergence
  - Wasted effort
    - Designers must spend significant time understanding what each tool supports and getting the constraints into each tool
    - EDA developers wind up defining new formats for each new tool
    - IP providers must supply the same data in multiple formats
    - IP integrators may have to translate internal constraints for IPs to get through their particular flow
    - Semiconductor vendors have a harder time qualifying tools

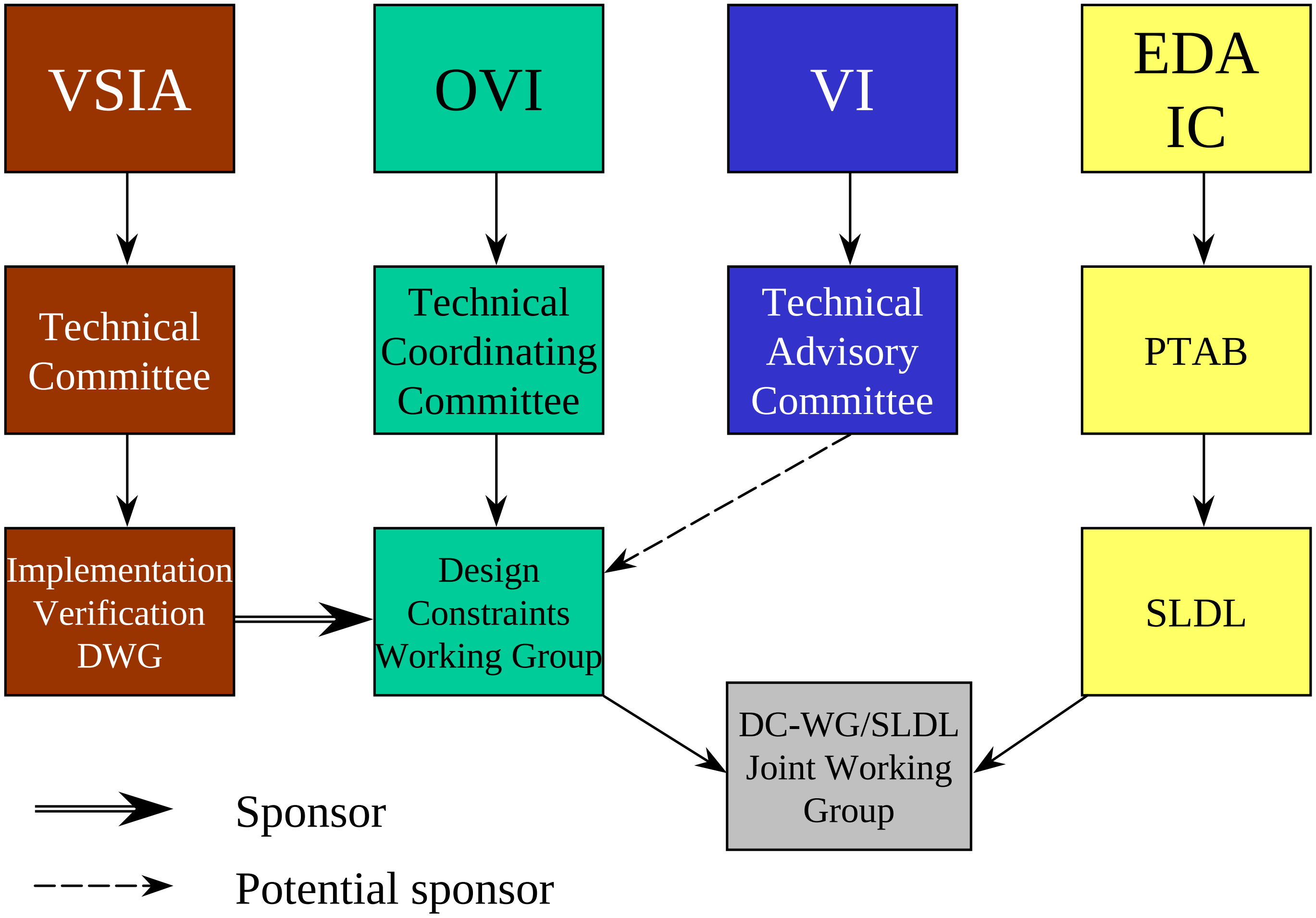
History

- Synthesis Constraints Working Group (SC-WG)
  - ◆ Formed in March, 1996 under OVI
  - ◆ Joint OVI/VI sponsorship in August, 1996
  - ◆ Charter
    - Synthesis tool interoperability
  - ◆ Focus
    - Definition of the General Constraint Language (GCL), a constraint command language for user entry
  - ◆ Problem
    - Consolidation of synthesis tools
  - ◆ Status
    - Fairly good progress on timing constraints
    - Inactive since October 1997
  - ◆ Details at http://www.eda.org/dcwg/scwg/index.html

History (2)

- DC-WG was formed in March, 1998
- Goals in starting a new effort:
  - Broader scope than SC-WG
    - Design constraints in general, rather than just synthesis constraints
    - Increased number of stakeholders => more participation
    - Avoid a win-lose outcome by addressing a wider range of issues than supported by any particular tool
  - Increased visibility and likelihood of adoption through VSI sponsorship
    - Cross-participation between VSI I/V DWG and DC-WG
    - Endorsement from VSI spurs interest from IP suppliers, IP integrators, and EDA vendors

Where Does DC-WG Fit?



VSIA

Technical Committee

Implementation Verification DWG

OVI

Technical Coordinating Committee

Design Constraints Working Group

VI

Technical Advisory Committee

EDA IC

PTAB

SLDL

DC-WG/SLDL Joint Working Group

Sponsor

Potential sponsor

VSIA Relationship

- Formal Sponsorship: VSIA will
  - Recruit members
  - Provide requirements specifically for IP mix and match
  - Endorse the standardization effort
    - Based on commitment to address VSIA requirements
  - Review draft specification, provide feedback
  - Adopt the standard when approved
    - Provided it meets VSIA requirements
  - Promote the standard after approval

SLDL Relationship

- Goal
  - Define syntax and semantics that can be used seamlessly from system level design through detailed implementation
- Approach
  - Joint Working Group
- Responsibilities
  - Define the general syntax and structure for DCDL
  - Define the conceptual model for constraints.
  - Define the formal information model for various constraint domains

DC-WG Charter

- Develop a constraint specification standard which captures aspects of the design intent besides logic functionality
- Constraints, assertions, and environment conditions
- Many types of constraint domains
  - Timing
  - Area
  - Clocking
  - Logic architecture
  - Power
  - Physical Implementation
  - Signal Integrity
  - Test
  - Environment/Operating Conditions
- Language independent (Verilog, VHDL, SLDL)

Deliverables

- Conceptual Model
  - How constraints are used
- Constraint Taxonomy
  - Short summary of types of constraints and semantics
  - Detailed, language-independent description of parameters and semantics
- Formal Specification
  - EXPRESS model
  - Currently expect this to be primarily for internal use
- Design Constraint Description Language (DCDL)
  - User entry, tool interchange mechanism
  - Examples
    - Ambit constraint commands
    - Synopsys dc_shell constraint commands

Details on DCDL

- DCDL is
  - A set of constraint primitives
  - Fully-defined semantics
  - TCL-compatible syntax, embeddable in SLDL Phase II
  - Name and value-based
    - Explicit values for constraints are applied to objects in the design that are identified by name
  - The result of expanding an application-specific TCL script
  - The expanded form used to interchange constraints between tools
- DCDL is not
  - An extension language
    - No general programming capabilities
  - Application or environment-specific
    - No macros to identify design objects or get values from library

Plans

- 1999
  - Release timing subset of DCDL
  - DAC ‘99 demos (synthesis, timing analysis, …)
  - Partial conceptual model
- 2000
  - Release power and signal integrity subsets
  - Full conceptual model
  - Formal EXPRESS model

Current Status

- Selected strawman for DCDL: Ambit Constraint Language
  - ◆ Balloting period closes Monday, 1/18, on vote whether to accept the strawman
    - Unanimously yes, as of 1/17
  - ◆ Covers timing, clock constraints
  - ◆ TCL-compatible
- DC-WG is free to use it as-is, build on it, modify it, or develop an alternative
  - ◆ Plan to build on it
- Also using Cadence’s GCF as a reference
- Constraint taxonomy is nearing completion

How Will DCDL Be Used?

- By designers
  - As a single, consistent basis for describing their intent
- By EDA tool developers
  - As a standard base for reading, writing, and interpreting constraints
- By IP providers
  - To describe their intent for partially implemented IP blocks
  - To describe restrictions on how IP blocks may be used
- By IP integrators
  - To complete the implementation of IP blocks
- By semiconductor vendors
  - As part of tool qualification
  - In creating design flows and kits

Contact Information

- Web Page
  - ◆ http://www.eda.org/dcwg
- E-Mail Reflector
  - ◆ dcwg@eda.org   (instructions for subscribing are on web site)
- Meetings
  - ◆ Teleconferences 9-11 am (PDT) every other Tuesday
  - ◆ Setting up face-to-face meeting at ICCAD

Backup slides

Synopsys Design Constraints

- Synopsys licenses the Design Compiler™ and PrimeTime™ constraint command languages
- OVI policy is to develop public-domain standards, based on a proven strawman where possible
- VSI preference is suitable *de facto* standards with no royalties or license fees
- Synopsys proposed DC-WG collect input and make suggestions for how to evolve the Synopsys formats
- DC-WG members felt that
  - Public-domain standards are preferable to licensed standards
  - The Design Compiler and PrimeTime constraint languages do not address many of the target domains
  - Conceptual model will address many unresolved issues, may affect language and semantics
  - Depending on evolution of a proprietary standard is troublesome

General Constraint Format (GCF)

- Reference for DC-WG
  - An exchange format for tool-to-tool communication
  - Cadence-proprietary format
  - Status
    - Initial emphasis on timing
    - Some area, power, parasitics, signal integrity constraints
    - Supported by many Cadence tools as well as Ambit
  - Plans
    - Continue to evolve to cover additional constraints
    - Consistent semantics with DCDL
    - Eventually replaced by DCDL