IEEE P1500 Core Test Description Language Task Force Report

Kenneth D. Wagner (S3, Inc.)
http://grouper.ieee.org/groups/1500/
Purpose of Standardization

Core Test Description Language (CTDL) Task Force

- **Current Members**
  - K.Wagner, B.Koenemann, P.Varma, S.Gupta, S.Yadavalli, R.Gupta, J.Rajski, E.Marinis, J.Beausang

- **Representing**
  - Semiconductor Manufacturers, EDA Vendors, Academic, System Vendors

Task Force Meetings

- Biweekly meetings in 2nd half 1997

Revised Draft (v0.6) in-process

- **Version 0.6**: *CTDL: A Representation Method and Description Language for Embedded Core Test Attributes and Data*
- **Version 0.4 4/25/97**: *Test Sockets: A Framework for System-On-Chip Designs*
Test Socket Basics

- The Test Socket provides an interface for specifying core test resources and operational modes

- The Test Socket information model should
  - Guide construction/checking of access features (e.g., digital/analog multiplexers, valid access path types and sensitization conditions, etc.)
  - Define connections to and configuration of Test Utilities (e.g., on-chip BIST controllers, Test Access Ports, etc.)
  - Link the Test Socket interface to structural model/support configurations for external/global test generation
  - Provide a reference interface for attached test data (e.g., signal names, test configurations)
  - Offer the ability to embed pre-defined test data
Anticipated Test Socket Features

- Multiple Configurations
  - e.g., internal test, external test, safe state, etc.

- Pin Definitions and Attributes
  - “Type” (e.g., timed/static signal, scan-in/out, don’t care, etc.)
  - “Stitching” (e.g., from PI, to TAP, timing constraints, etc.)
  - “Custom” (e.g., BIST configuration data, etc.)

- Links to Models for Chip Integration
  - e.g., “skin” model (for external test), full model (for re-generation of internal tests), etc.
Basic concepts and subset of features
  - Driven by simple examples

Combination of structural model and Test Socket Description Language
  - Draft uses HDL-neutral pseudo-language

Example features include basic ability to
  - Apply pre-defined tests to embedded macros
  - Protect macros while other parts of chip are tested
  - Generate chip-level tests for functions surrounding the macros
Control - Operational Protocols
  - Initialization including transparency
  - Test Pattern Delivery
  - Separately specify test socket structure/architecture from core test operation requirements

Transparency [ Full / Partial ]
  - Combinational & sequential transparency
    • input, output (incl. conditioning) and power isolation

Core Test Data
  - Referenced but not contained in CTDL

System Chip Test Description Language (SCTDL)
  - Adjunct initiative
  - To assist the system integrator in developing the system-level test (describes system-level access)
Core Test Information Model v0.6

core test model file(s)

block definition file(s)

CTL file(s)

signal x  signal y

port a  port b

Block Instance

core test model statements
(e.g., gate level or RTL
Verilog/VHDL/EDIF)

IO port declarations

CTL file reference

model statements
(e.g., gate level or RTL
Verilog/VHDL/EDIF)

CTL statements
- structure/operation
Other Version 0.6 Changes

- **CTDL file**
  - Referenced from VHDL, Verilog or EDIF core description
  - Pattern on STIL (P1450) format as much as possible
    - protocols, simple structures, timing information, test data

- **Defined Keywords**
  - Basic Configurations
  - Support for timed Probe pintype requirements
    - Timing Core Pin to/from Chip Pin
      - min/max latency, jitter, max skew pin group to reference pin

- **Pin Type extensions**
  - Scan types, bidis
Version 0.6 Limitations

- Intermode Dependencies
  - work required

- Hierarchy
  - 2-level support only
  - Recursion for hierarchical support

- Conditional Operation
  - not supported

- Not Diagnostics
  - not yet considered

- No architecture specific configurations

- No analog or mixed signal
Next Steps

- **CTDL Task Force**
  - Commitment to add 5 new members incl. new co-Chairman
    - Representation for ATE vendors, STIL
- **Restart Task Force Meetings**
  - ASAP
- **Complete Version 0.6**
  - *CTDL: A Representation Method and Description Language for Embedded Core Test Attributes and Data*
- **Maintain revisions**
  - More careful maintenance

**VOLUNTEERS NEEDED AND WANTED!**
APPENDIX: Core Basics

- Cores are:
  - Pre-designed or modified functional entities
    - by core-developers/designers,
    - encapsulate intellectual property for re-use;
  - For embedding into semiconductor chips
    - by core-users (often not core-developers),
    - with other cores and user-defined functions;
  - Using (deep) submicron technologies
    - by leading-edge semiconductor vendors (can be other than core-developers or core-users),
    - exploit system-level integration capabilities.
APPENDIX: Cores Are Shared Responsibility
APPENDIX: Types of Cores

■ Mergeable Core:
  - pre-designed for integration with user-defined functions for common implementation flow,
  - becomes part of the user-defined function,
  - is tested as part of the user-defined function.

■ Non-Mergeable Core:
  - pre-designed for cooperation with user-defined functions,
  - remains separate entity that interfaces with the user-defined function,
  - has separate test requirements or test data.
APPENDIX: Core Test Issues

Key Issues to Address Are:
- Core Preparation
- Core Packaging
- Chip Assembly
- Chip Test Generation
- Chip Test Verification and Debug
- Board/System-Level Test Support

The Industry Needs a Comprehensive Framework and Support Capabilities
- Methodology Definition(s)
- Test/Testability Information Exchange Models
- Tools and Services