Status Report

P1500 Compliance Definition Task Force

Erik Jan Marinissen
Philips Research Laboratories
Eindhoven, The Netherlands

IEEE P1500 WG Meeting
International Test Conference (ITC'02)
October 7, 2002 - Baltimore, MD, USA

Task Force Organization

- **Mission**
  - Definition of two compliance levels: ‘1500-Unwrapped’ and ‘1500-Wrapped’
  - Guaranteeing interoperability to both core provider and core user in use scenarios
  - Write Compliance Definition descriptions to go into draft standard
- **Members**
  - Karim Arabi (PMC-Sierra)
  - Rohit Kapur (Synopsys)
  - Brion Keller (Cadence)
  - Erik Jan Marinissen* (Philips)
  - Jon Udell (Mentor Graphics)
  - Yervant Zorian (LogicVision)
- **Meetings**
  - Weekly teleconference meetings of 1 hour – Wednesday 9-10 h PST
    Thanks to IBM (Bernd/Brion) for providing facilities!
  - Password-protected web site at IEEE computer *(old LTF web page)*

Dual Compliance Concept

(repeated from previous CD-TF Status Reports)

- **IEEE 1500 Unwrapped**
  - Core which does not have a complete IEEE 1500 wrapper, but does have an IEEE 1500 CTL description on the basis on which the core could (‘easily’) be made ‘IEEE 1500 Wrapped’ (either manually or automatically by tools)
  - Complete IEEE 1500 CTL description describing how to test the core

- **IEEE 1500 Wrapped**
  - Incorporates complete IEEE 1500 wrapper function
  - Complete IEEE 1500 CTL description describing how to test the core (including how to operate the wrapper)

General Rules

- All test information of the core shall be provided in CTL.
- As relevant on a per-signal basis every CTL statement in the Internal block of statements shall be used for the test modes described for the core.

Per-Signal Rules

- All signals of the core should be identified using the Signals block of statements in CTL.
- All non-digital signals identified for the core should be classified according to their Electrical Characteristics using the following statement.

```
   signame { ElectricalProperty property_type; }
```

- All digital signals of the core should be categorized according to their test function for all test modes using the following CTL statement.

```
   signame { DataType (data_type)+; }
```

Other Rules – 1

- Active states of test mode signals and scan enable signals as needed for the validity of test information of the core should be identified using the ActiveState statement associated with the DataType.

```
signame { DataType data_type { ActiveState active_state; } }
```

- Certain signals such as clocks, test-mode-signals and Set, Reset and Clear signals are assumed to be at a certain state at the beginning of every test protocol for sequences to be valid. This state is required to be specified using the AssumedInitialState statement associated with the DataType.

```
signame { DataType data_type { AssumedInitialState assumed_state; } }
```

- If the state of the core relies on stability of certain core-input-signals during the scan operation of the embedded environment this information should be specified with the InputProperty.

```
signame { InputProperty ScanStable; }
```

Other Rules – 2

- Every core should come with at least one definition of the Internal test mode of the core in CTL. If the logic model of the core is not available then the test patterns should be provided using CTL.
- All state elements of a core that are part of the final wrapper implementation of the core should be described in CTL as part of a scan chain using the ScanStructures construct of CTL.
- If a state element exists in internal to the core that is to be reused as part of the wrapper it should be described in CTL using the following statement.

  ```
  signame { IsConnected { StateElement Scan cellname; } }
  ```

- If a IEEE1500 cell exists internal to the core it should be identified using the following statement.

  ```
  signame { Wrapper IEEE1500 CellID 1500_cell_names; }
  ```

- All digital signals of IEEE1500 wrapped core with no internal wrapper scan cell should be identified in CTL using the following statement.

  ```
  signame { Wrapper None; }
  ```

Other Rules – 3

- All IEEE 1500 signals that are used to operate the wrapper are to be identified with the following statement.

```
signame { Wrapper IEEE1500 PinID ; }
```

- Test patterns shall use test protocols in CTL such that the protocol does not assume that consecutive test patterns are overlapped. Such protocols are identified identifiable in CTL with the DoTest keyword.
- Safe mode and associated safe values. A safe mode is required to be described for Un-Wrapped cores.

Other Rules – 4

- Identification of events in the protocol that operates the WIR. The Capture, Shift and Update events of the protocol that operates the WIR should be identified in CTL for the wrapped core using the Identifiers syntax or as a purpose of the protocols. A typical integration of wrapped cores would daisy chain the WIRs of all the wrapped cores. To allow for this form of connection the activities of the WIRs need to be synchronized. The information is required to perform this task.
- All bi-directional, differential core I/Os or any other I/Os that have a chip pad implemented internal to the core shall have wrapper cells built in.

Wrapper Parameters

- **Bandwidth**
  - Number of WPI-WPO pairs (zero or more)
  - Width of each WPI-WPO pair (if present)
- **Instructions**
  - Optional instructions
  - User-defined instructions
  - OpCodes of instructions

- **WBR Functionality**
  - Shared or dedicated wrapper cells
  - Shift-Only or Shift+Update wrapper cells
  - Storage capacity (one or more bits)
  - Location of capture (in Shift or Update register)
  - Ripple protection (with Update register or gate)
  - ‘Safe’ output values

Ongoing Activities

- Going through entire IEEE P1450.6 (CTL) document
  - Drafting rules for all CTL key words that have impact on wrapper design and/or wrapper operation?
- At same time, trying to verify whether CTL currently meets the P1500 / CTAG / wrapper design/operation needs

Ongoing Activities – 2

- “TimingCritical” ⇒ “TimingSensitive” (default: non-sensitive)
- Patterns
  - Distinguish between Verification and Manufacturing patterns
  - Fault model and fault coverage required?
  - Do compliant cores need to come with at least one pattern?
  - “Foreign” patterns
    - Not every STIL test pattern set is CTL compliant (e.g., separation of test protocol and test data)
    - STIL patterns might be “Foreign” to CTL!
    - P1500 should exclude the “Foreign” patterns in CTL

Again: Compliance Level Naming

- **P1500-Ready / P1500-Compliant** (Original)
  - ‘P1500’ is not good, ‘P’ will eventually drop
  - ‘P1500-Ready’ is also a compliance level
  - ‘Ready’ seems to indicate I am done, whereas I am just halfway
- **Level-1 Compliant / Level-2 Compliant** (P1500/D0.1)
  - No semantics attached to ‘numbers only’
- **1500-Unwrapped / 1500-Wrapped** (VTS’00)
  - ‘Unwrapped’ might indicate that a once existing wrapper was removed
  - ‘Unwrapped’ might have a (non-P1500) wrapper too
  - Naming focus too much on wrapper only

- New suggestions
  - **1500-Wrappable / 1500-Wrapped**
  - **1500-Prepared / 1500-Wrapped**