

## **Status Report**

# **P1500 Compliance Definition and Information Model Task Force**

**Karim Arabi**

**PMC-Sierra, Inc.**

Presented at IEEE P1500 WG Meeting

VLSI Test Symposium 2003

Napa Valley, California, USA

April 27, 2003

---

# Task Force Organization

- Current Members
  - Karim Arabi\* (PMC-Sierra)
  - Rohit Kapur (Synopsys)
  - Brion Keller (Cadence)
  - Jon Udell (Mentor Graphics)
  - Yervant Zorian (Virage Logic)

# Task Force Organization

- Mission
  - Definition of two compliance levels: “1500-Wrapped” and “1500-Unwrapped”
  - Definition of information model for “1500-Unwrapped” and “1500-Wrapped” cores
  - Guaranteeing interoperability to both core provider and core user in different use scenarios
- Meeting
  - Weekly teleconference meetings of 1 hour on Wednesdays from 9 AM to 10 AM PST
    - Thanks to Cadence/IBM (Brion) for providing the phone services
  - Materials archived at private web site

# Dual Compliance Concept

- “IEEE 1500 Unwrapped” Core
  - Does not have a complete IEEE 1500 wrapper
  - Has an information model based on CTL and compliance rules
    - Can be used to convert the “IEEE 1500 Unwrapped” core to an “IEEE 1500 Wrapped” core either manually or automatically using tools
- “IEEE 1500 Wrapped Core”
  - Incorporates complete IEEE 1500 wrapper function
  - Has an information model based on CTL and compliance rules
    - Can be used to integrate the core at the top-level and generate vectors for the top-level interconnect
    - Can be used to convert the “IEEE 1500 Wrapped” core to an “IEEE 1500 Unwrapped” core either manually or automatically using tools

# Schedule

- Release the initial draft of compliance chapters to CTAG and Document teams on April 2 <Done>
- Finalize the compliance chapters on May 15, 2003
- Integrate compliance chapters in version D07 in June 2003
- Reintegrate the Compliance Definition and Information Model Task Force within the 1500 working group in June

# **IEEE 1500 Compliance Rules**

---

# Common Rules: Test Mode Information

- All test information regarding the core shall be provided in CTL. This information should not rely on user specific extensions or functions such as STIL extensions.

```
Environment {  
    CTL config1 {  
        TestMode InternalTest { AlternateTestMode config2; }  
    }  
    CTL config2 {  
        TestMode InternalTest { AlternateTestMode config1; }  
    }  
    CTL config3 {  
        TestMode ExternalTest;  
    }  
}
```

# Common Rules: Per-Signal Information

- 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;  
}
```

- Unless specified the property type of a signal is assumed to be digital since IEEE 1500 is targeted for digital signals.
- CTL allows for the description of property types that include digital, analog, power and ground signals.

# Common Rules: Per-Signal Information

- 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)+;  
}
```

- scan related signals are to be identified using the **DataType** statement in CTL for all the test modes required for the associated compliancy level.

# Common Rules: Test Pattern Information

- All test patterns must be supplied using the CTL dialect of the STIL test pattern language. This specifically excludes the use of the foreign pattern construct.
- It is recommended that all timing information in the CTL **DriveRequirements** and **StrobeRequirements** blocks be specified with acceptable margin to allow maximum flexibility for event placement.
- Test patterns shall use test protocols (Macros or Procedures) in CTL such that the protocol does not assume that consecutive test patterns are (scan) overlapped. Such protocols are identified in CTL with the **DoTest** keyword.

# Common Rules: Test Pattern Information

- Test setup patterns should be separated from the test data patterns.
- There should be a sufficient set of patterns supplied with the core to ensure that every test setup and every protocol (Macro and Procedure) is used at least once.
- If the full internal test pattern set is not supplied with the core (to the core integrator), the complete pattern set must be made available to the device/chip manufacturer or testing company to use during production test.

# Common Rules: Test Pattern Information

- Fault grading results for all internal patterns and/or pattern\_bursts should be provided. This data, in CTL format, includes the total number of faults, the number of faults detected, the number of redundant faults, the number of ATPG untestable faults and the number of possibly tested faults.

# Unwrapped Core Rules: Test Logic Information

- All core pins should be described using CTL to define the type of wrapper cell needed for each core pin.
- All bi-directional and differential core I/Os shall have wrapper cells built in.
- Any core I/Os that have a chip pad implemented internal to the core shall have wrapper cells built in.

# Unwrapped Core Rules: Test Logic Information

- If the unwrapped core has some functional I/Os that are registered and the registration is desired to be shared with core wrapper cells,
  - The I/O registration must be done with cells that are IEEE 1500 compliant wrapper cells.
  - Boundary cells must be connected within scan chains containing only such wrapper cells.
  - Such scan chains can consist of single boundary cells or may contain several wrapper cells. Full control of all such wrapper cell scan chains must be available from core I/O pins.

# Unwrapped Core Rules: Test Logic Information

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

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

- 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.

# Unwrapped Core Rules: Per-Signal Information

- 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; }  
}
```

# Unwrapped Core Rules: Per-Signal Information

- 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**.

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

# Unwrapped Core Rules: Safe/Stable State Information

- 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;  
}
```

- One example is a core with programmable clock generator that may generate out of range frequencies if the core inputs are not stable.

# Unwrapped Core Rules: Safe/Stable State Information

- If the state of the core environment relies on stability of certain core output signals during the scan operation of the core this information should be specified.
  - One example is a core with tri-state outputs that may be placed in hiz state when the core is in scan test mode.

# Wrapped Core Rules: General Information

- All information related to the test wrapper must be supplied in the CTL format if it supports the definition of such information.
- All wrapper supported public instructions should be described in CTL.
- All IEEE 1500 wrapper pins should be described in CTL.
- Relevant input and output safe states should be provided for all possible test operating modes, including INTEST and EXTEST.

# Wrapped Core Rules: General Information

- 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.
  - Required to synchronize the activities of WIRs from different cores

# Wrapped Core Rules: Per-Signal Information

- 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;  
}
```

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

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

# Wrapped Core Rules: Test Logic Information

- 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 internal to the core that is reused as part of the wrapper it should be described in CTL using the following statement.

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

- Facilitates conversion to “1500 Unwrapped” core