P1500 CTL
Proposing a Standard Core Test Language

Brion Keller
kellerbl@us.ibm.com

Rohit Kapur     Maurice Lousberg     Tony Taylor

Prab Varma      Paul Reuter         Ted Eaton
P1500

- P1500 strives to provide a standard means to test many types of embedded Cores
  - Hard Cores primarily
  - Both “Black Box” and Mergeable Types of Cores
  - With or without Core shell to aid testing
  - Guidelines for Core DFT and how to embedd Cores yet keep them testable
P1500 Scaleable Architecture

- Scaleable Architecture Group (aka CTAG) defines DFT for cores such that
  - A Wrapper can be added to a core that will make testing the core and the surrounding logic easier
  - There is a standard way of designing wrappers
  - Wrapper may be able to communicate to on-chip test-access controller which controls which core is being tested
P1500 CTL

- P1500 CTL strives to provide a standard means to describe the test requirements of embedded Cores
  - Hard Cores as Black Boxes is the primary target
  - With or without Core (boundary) shell to aid testing (will work with P1500 Wrapper group)
  - CTL describes requirements on surrounding logic needed to efficiently test the core
CTL Can be Used for...

• Helping to insert a Core into surrounding logic
• describing how a synthesized boundary shell works
• Telling ATPG software how the core pins need to be accessed in order to be able to map the core test patterns to the chip boundary
• Passing along the actual core test patterns (in STIL format)
A Core Test Language

Chip

Core

This is the way to scan
This is the timing requirement
This is an assumption
This is what you do to get into quiet mode
This is what you do to burn the chip...
Progress Report

• Come a long way since kick-off (12 May 98)
• 23 meetings to date.
• Typical attendance per meeting (4/5).
• Defined our scope (In or Out).
• March 30th we completed the 1st Pass.
• Cleanup phase has started.
Scope of CTL

• Identify the requirements of the Test Data.
• Identify the portions of the Test Data relevant to the system integrator.
• Define configurations of the core.

To Enable … Testing the Core & UDL
To Enable … Use of the Core Test Wrapper
Summarizing CTL Activities
Giving the Monkey Some Character

Giving a meaning to the 1’s and 0’s
CTL is to be based on STIL

- STIL Language extensions needed to support System Integration needs.
- STIL structure modifications to allow for a Core Test Language (Multiple Test Mode needs).
- Substantial set of new signal Attributes to define core pins’ function and needs.
Information in CTL can be used for different SoC solutions
CTL by Example (Scan Chain)

**Header** { Title “scan example”; Date “VTS99”}

**Signals** { cSI In {ScanIn 6;}
             cSG In;
             cSO Out {ScanOut 6;}}

**ScanStructures** {
  ScanChain internalScanChainC {
    ScanLength 6;
    ScanInversion 0;
    ScanCells {c0 c1 c2 c3 c4 c5;}
    ScanIn cSI;
    ScanOut cSO;
    ScanMasterClock cClk;}
}

**MacroDefs** {
  “cChainOperation” {
    Purpose ControlObserve;
    W cChainTiming;
    C { cSG = 1;}
    Shift { V { cSI=#; cSO=#; cClk=P;}}
  }
}
P1500 Ready for CTAG

Header { Title “P1500-Ready-CTAG”; Date “VTS99”}
Signals {
    a[0..5] In {
        CoreInternal {DriveAccuracy NonCritical;
                      DriveChar Transition;
                      DataRate 1;
                      CaptureClock PositiveEdge cClk;}}
    b[0..5] Out {
        CoreExternal {StrobeAccuracy Critical;
                      ConnectTo PO;}
        CoreInternal {StrobeChar ScanStable;
                      DriveChar SkewTolerant;}}
    TestMode In {
        CoreInternal {DataRate 0;}}
    cClk In {
        CoreInternal{DataChar Clock;
                     DriveAccuracy Critical;}}
}
**P1500-Compliant Wrapper Inside**

**Header** { Title “P1500-Compliant”; Date “VTS99”}

**Signals** {
    a4 In {
        CoreInternal {
            ConnectIn internalScanChainC, 5, 0;
        }
    }
    b4 Out {
        CoreInternal {
            ConnectOut internalScanChainC, 8, 0;
            EnableOut CoreSignal Outen, ForceUp;
        }
    }
}


Header \{ Title "Isolate"; Date "VTS99";\}

PatternExec \{
  Timing Core;
  PatternBurst SetCoreToHighZ;
\}

PatternBurst SetCoreToHighZ \{
  CoreSetup \{
    Purpose Configuration;
    TestMode Isolate;
  \}
  PatList \{ configureCore; \}
\}

Pattern configureCore \{
  V \{ Outen = 0; \}
\}
CTL describes...

- Core Transparency (Possible Instruction in CTAG).
- Quiet (Iddq) configuration (Possible Instruction in CTAG).
- InTest (Possible Instruction in CTAG).
- ExTest (Possible Instruction in CTAG).
- Diagnostic/Production/Characterization
- PreAmble/PostAmble/Filler/Result
- BSDL-like info used to describe boundary scan shell if it exists and is not visible in netlist
Our Targets

• June freeze of functionality.
• October 1st draft.
• December final version for Ballot.