

# **SOC Testing**

- **SOC Test Problems/requirements**
- **IEEE P1500 Standard**
- **SOC Test Methodology**
- **Testable SOC Design Flow**
- **Conclusions**

# SOC Test Problems

- **Deeply embedded cores**
- **More, higher-performance core pins than SOC pins**
- **External ATE inefficiency**
- **Mixing technologies: logic, processor, memory, analog components**
- **Multiple hardware description levels for cores**
- **Different core providers and SOC test developers**
- **Core/test reuse**
- **Hierarchical core reuse**
- **IP protection**

# SOC Test Requirements

- Deeply embedded cores
  - Need Test Access Mechanism
- More, higher-performance core pins than SOC pins
  - Need on-chip, at-speed testing
- External ATE inefficiency
  - Need “on-chip ATE”
- Mixing technologies: logic, processor, memory, analog components
  - Need various DFT/BIST/ techniques

# SOC Test Requirements (cont.)

- Multiple hardware description level for cores
  - Need to insert DFT/BIST at various levels
- Different core providers and SOC test developers
  - Need standard for test integration
- Core/test reuse
  - Need plug-and-play test mechanism
- Hierarchical core reuse
  - Need hierarchical test management
- IP protection
  - Need core test standard/document

# Core Test Techniques

- **Single scan**
- **Multiple scan**
- **Broadcast scan**
- **Enabled ATPG – Scan insertion**
- **Reusable ATPG – Access & isolation**
- **Test point insertion**
- **Shadow register**
- **Enabled BIST – Scan, test points**
- **Embedded BIST – Serial or parallel, local controller, TPG and SA**
- **Boundary scan (BS)**

# Component test

- **DSP/CPU cores: BS supporting BIST, Scan, test point, shadow register.**
- **ASIC cores: BIST, Scan, shadow register, w/wo BS.**
- **Memory: Embedded BIST**
- **Analog: Test points, DSP, BIST, *ad hoc***

# IEEE P1500

- **Goals**
- **Task Force**
- **Basic Principles**
- **Overall Architecture**
- **Core Test Requirement / Architecture**
- **Wrapper Register Function / Configuration**
- **Wrapper Cells**

# Goals of IEEE P1500

## Standardize a Core Test Architecture which:

- Defines a **core test interface** between an embedded core and the system chip.
- Facilitate **test reuse** for embedded cores through core access and isolation mechanisms
- Provide testability for system chip **interconnect and logic**.
- Facilitates core test interoperability, with **plug-and-play protocols**, to improve the efficiency of test .



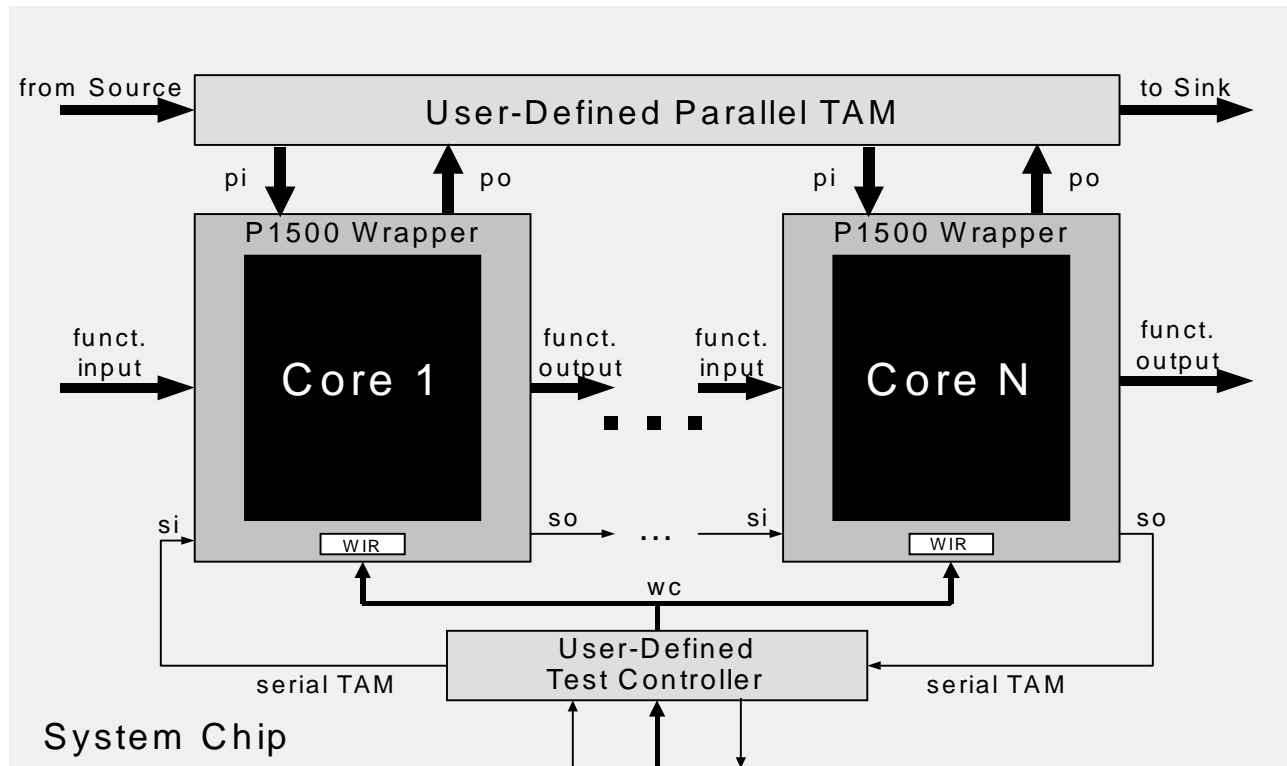
# Active Task Force for P1500

- **Advantest**
- **ASC**
- **G2Startup.Com**
- **HP**
- **IBM**
- **Intellitech**
- **LogicVision**
- **Mentor Graphics**
- **Motorola**
- **Nortel**
- **Palmchip**
- **Philips**
- **Sisco Systems**
- **Sonic**
- **Synopsys**
- **TI**
- **Veritable**

# Basic Principles

- **Embedded core test requires the following hardware components:**
  - **A Wrapper (around the core)**
  - **A Source/Sink for test patterns (on or off-chip)**
  - **An on-chip Test Access Mechanism (TAM) to connect the Wrapper to the Source/Sink.**
- **Faciliate test reuse for "non-merged" cores.**
- **Define the behavior of a standard Wrapper per core and its interface with a Test Access Mechanism (TAM).**

# Current Proposed P1500 Overall Architecture



Overview of the P1500 scalable architecture.

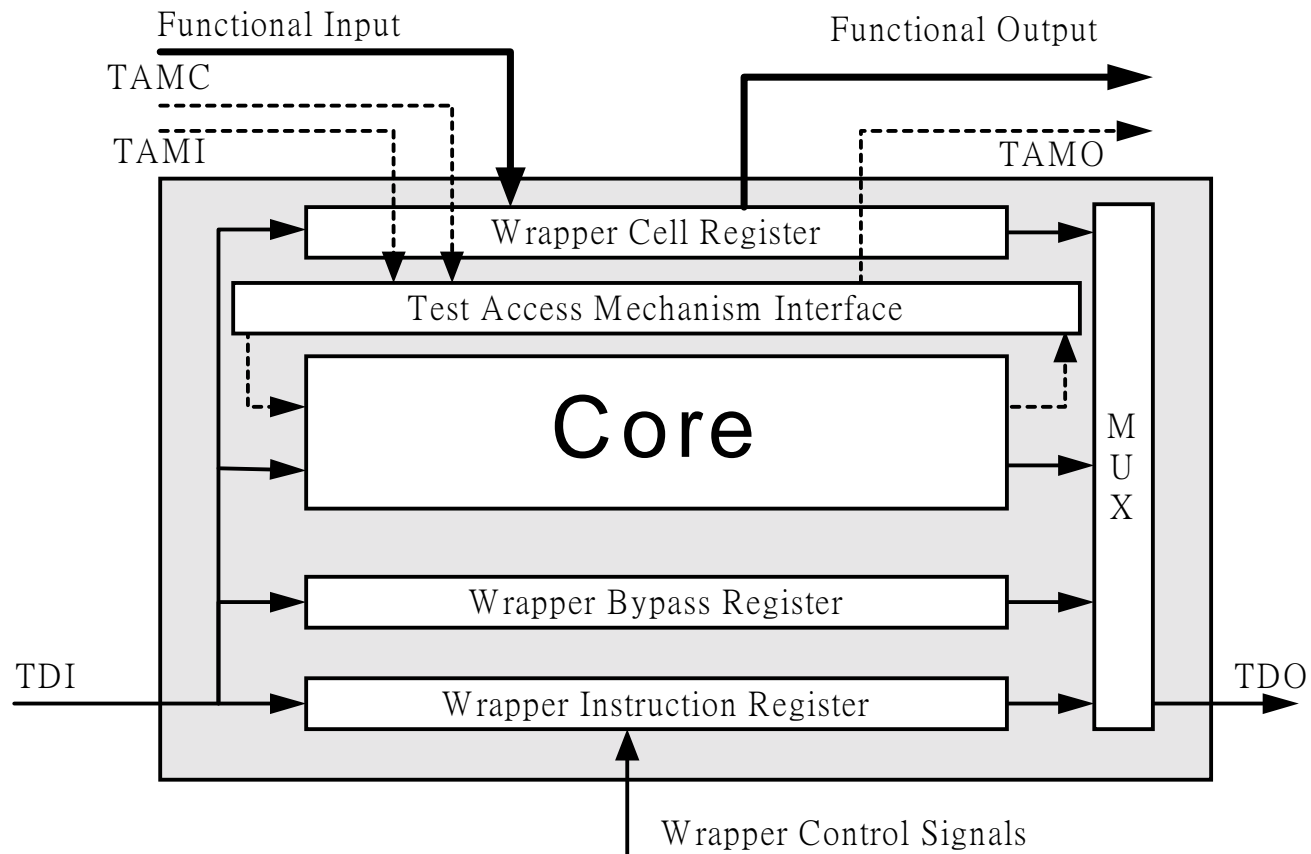
# Basic Principles (cont.)

- **Core test wrapper modes:**
  - Core Normal Mode
  - Core Test Mode (internal)
  - Core Interconnect Test Mode (external)
  - Core Isolation Mode
- **The standard Wrapper behavior may be:**
  - Implemented and provided by core vendors
  - Added to the core during a subsequent design stage. It is assumed that EDA vendors will:
    - offer tools to implement the standard Wrappers
    - check for compliance
    - provide system-chip level optimization.

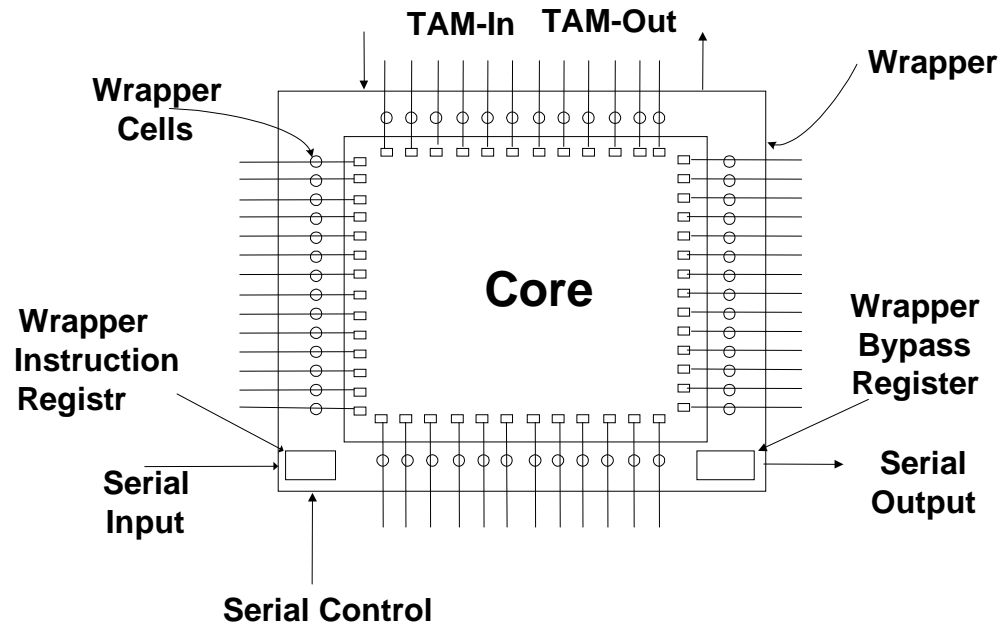
# Basic Principles (cont.)

- **Should standardize the interface between the Wrapper and Test Access Mechanism, but not the Test Access Mechanism itself of the system-chip.**
- **Will define how both 1149.1 & non-1149.1 cores can **co-operate** during test.**
- **Should specify the standard Wrapper and the interface to the Test Access Mechanism using the P1500 Core Test Language (CTL).**
- **Should be applicable to hierarchical cores.**

# A P1500-wrapped Core



# P1500 Architecture Components



## ❑ A P1500 wrapper contains the following:

- **A wrapper Instruction Register** for providing wrapper mode control
- **Wrapper Cells** to provide test functions at the core terminals
- **An optional Bypass register** for a single bit scan bypass through the wrapper
- **A serial interface** for providing initialization and communication to the Wrapper Instruction Register, Wrapper Cells, and Bypass register

# Required Modes for Embedded Core Test

- **Core Normal Mode**
  - Wrapper is transparent, core functions normally
- **Core Test Access Mode**
  - Core wrapper provides for controlling core inputs and observing core outputs during core test application.
  - Test Access Mechanisms (e.g., Test Bus, Test Rail, ...Other) configured during System Chip integration.
- **Interconnect & UDL Test Access Mode**
  - Core wrapper provides test observation at core inputs and control at core outputs.
- **Test Isolation Mode**
  - Not always required for every core or in every application
  - Can be achieved by constraining core inputs/outputs
  - Protects core and system chip from damage
  - Useful for reducing power consumption, and for Iddq testing



# Test Function at Core Terminals

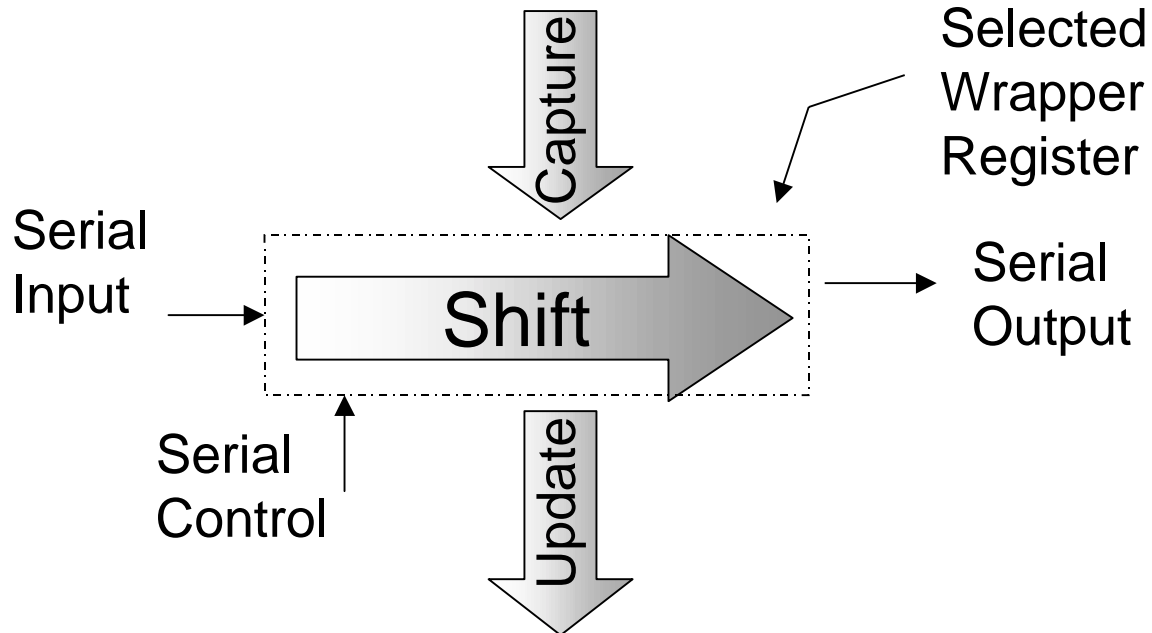
- **Input test functions**

- **Input observation:** for observation of external signal
- **Input control:** for applying test to core input
- **Input constraint:** to fix logic at core input

- **Output test functions**

- **Output observation:** for observation of internal core
- **Output control:** for interconnect test
- **Output constraint:** to fix logic at non-tristate output
- **Output disable:** for tristate driver

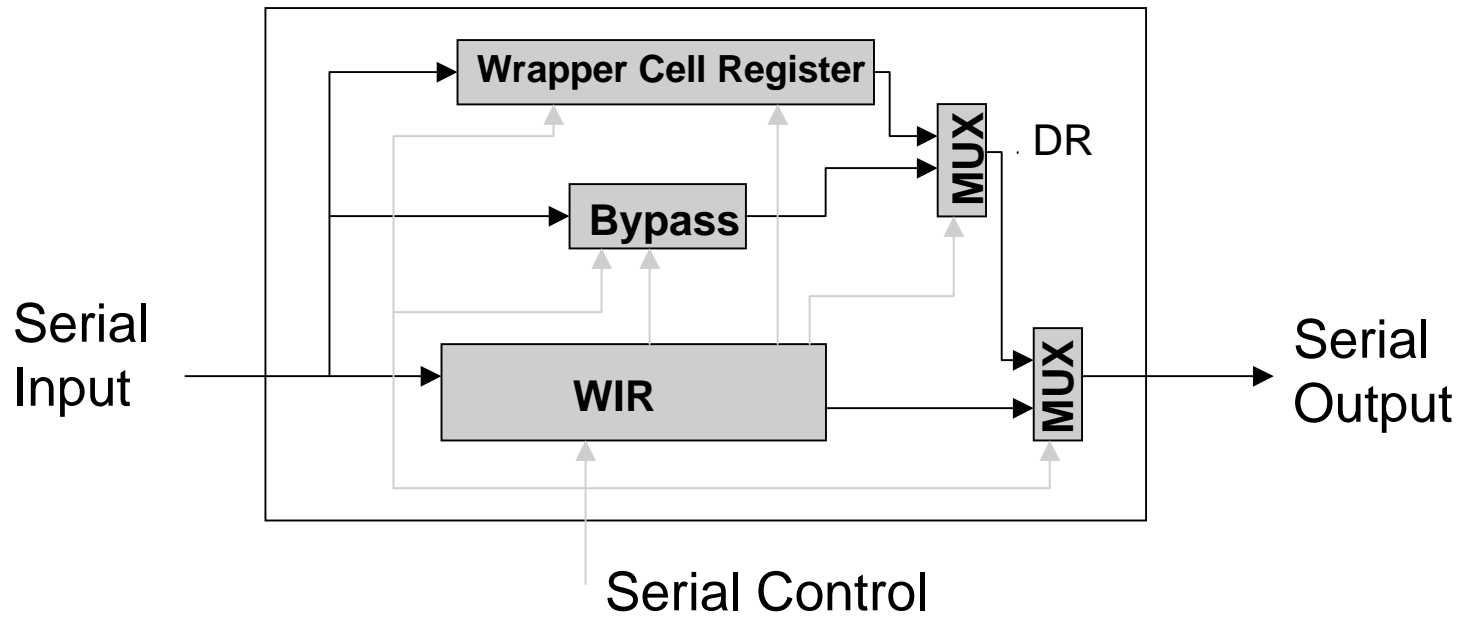
# Scan Protocol Behavior of Wrapper Registers



**Standard P1500 protocol for Wrapper Register will provide for:**

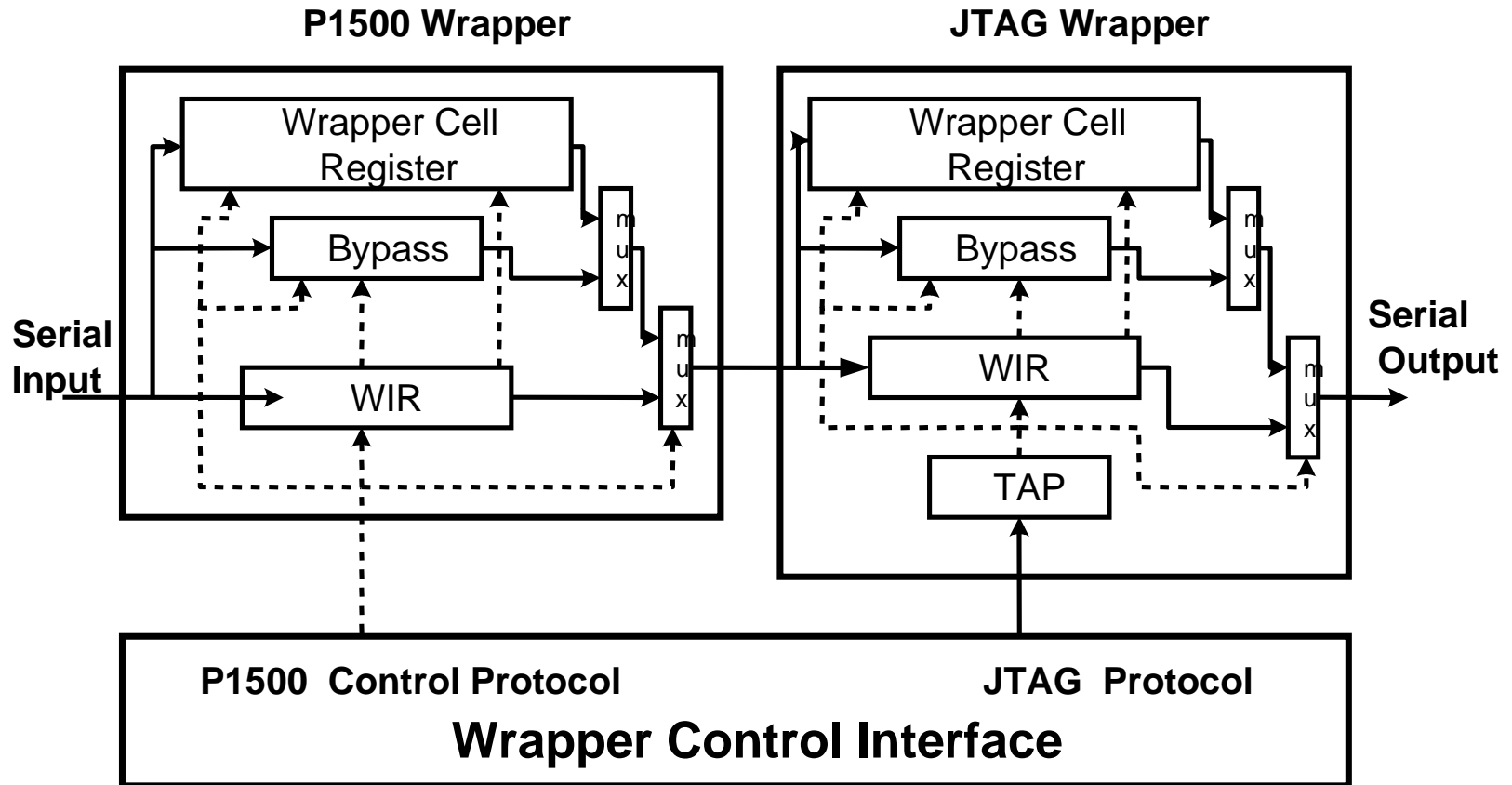
- **Parallel capture of input data into the selected register**
- **Serial shift of the register from serial input to serial output**
- **Update scan-in data of register to a parallel update stage**

# Standard Serial Scan Path Configuration



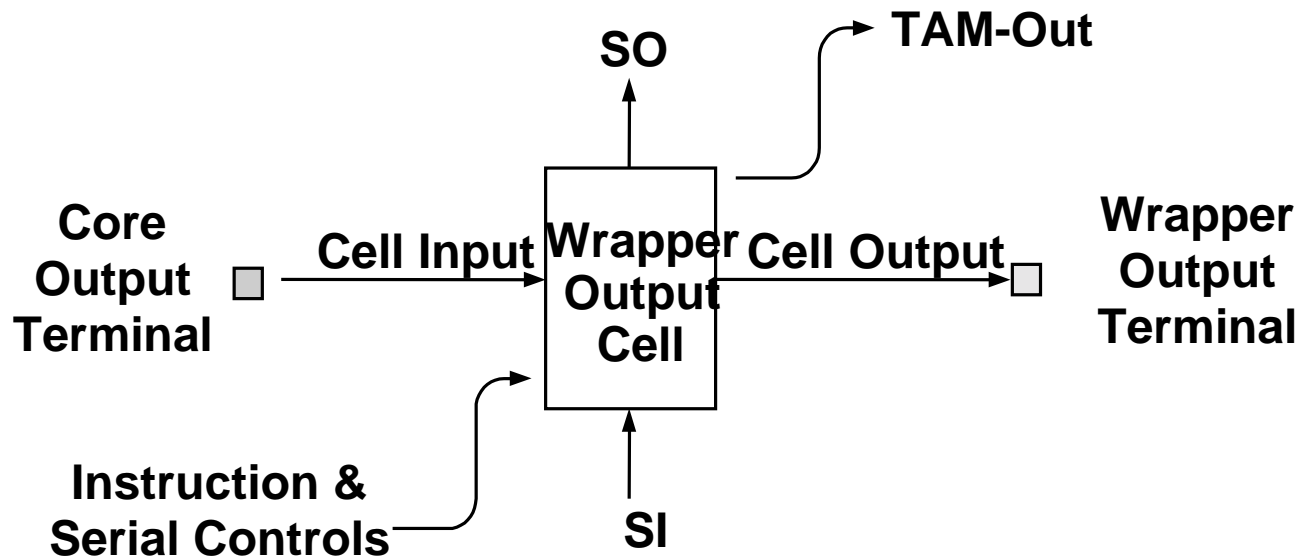
- ❑ Serial control lines enable & perform scan , and select between:
  - Wrapper instruction register (WIR)
  - Or other data registers (DRs), e.g. Wrapper cell register, bypass, etc.
- ❑ Updated WIR then selects between DRs
- ❑ Core test 1-N instructions permit TAM connection & configuration of Wrapper DRs, or internal core registers, to be *user defined!*

# P1500 Wrapper Connection



- ❑ Wrapper Control Interface is configured by system chip integrator
- ❑ P1500 & JTAG inter-operate at wrapper & serial data interfaces

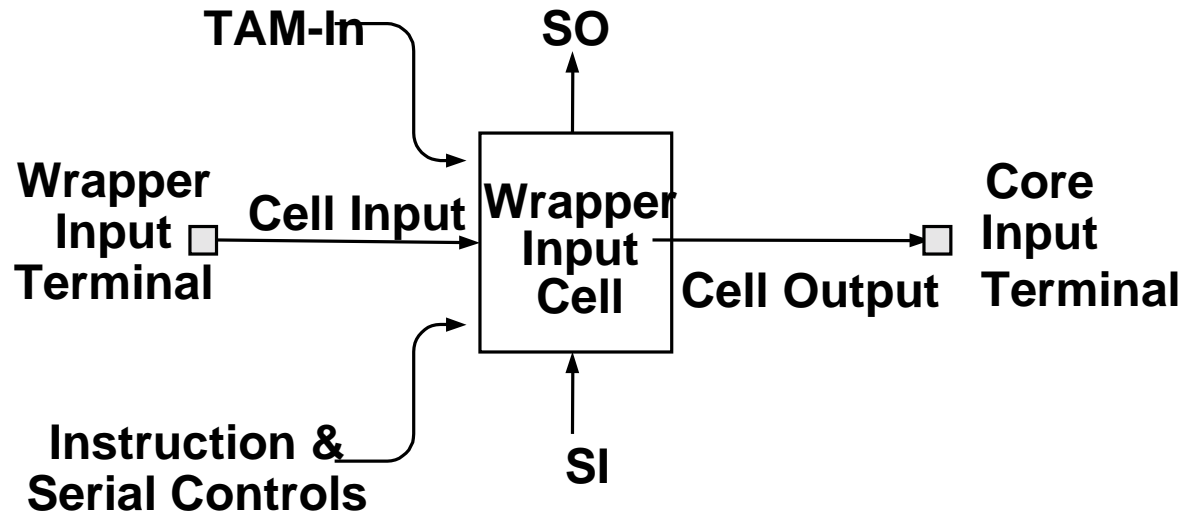
# Wrapper Cell Example--Dedicated Output Cell with Update Stage & TAM-Out



## Cell behavior for Wrapper Scan Protocol

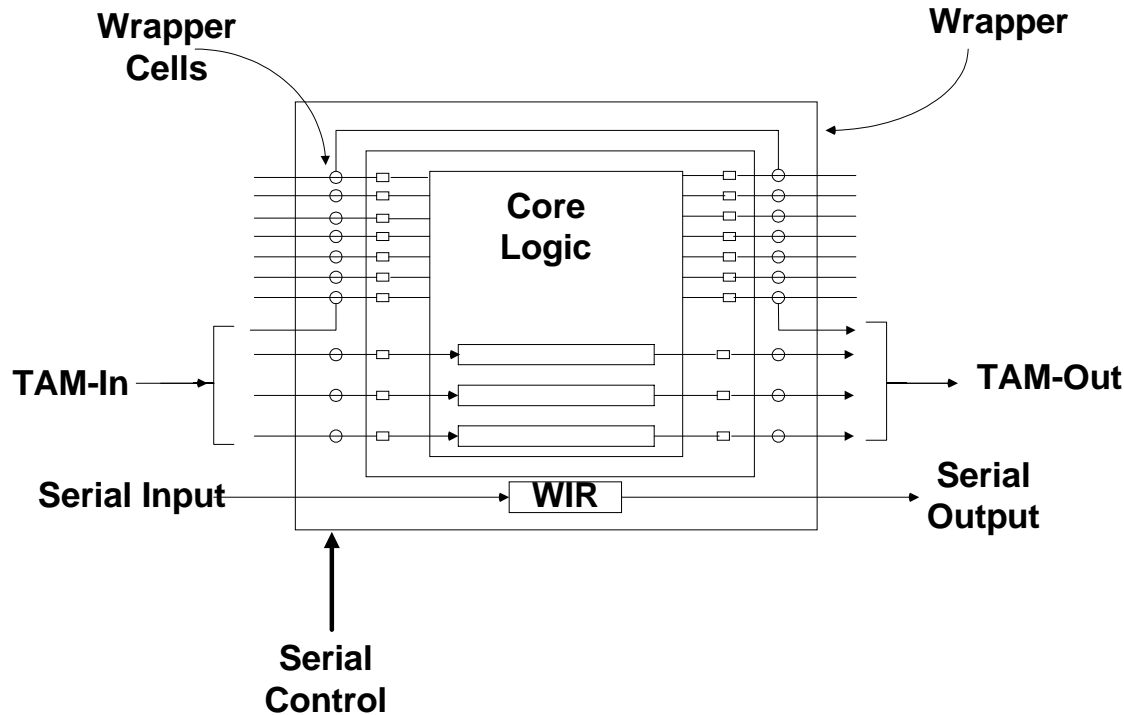
- Captures data at cell input
- Shifts data from scan input (SI) to scan output (SO)
- Updates shift stage data to update stage

# P1500 Wrapper Cell Example -- Dedicated Output Cell with Update Stage & TAM-In



- **Cell behavior for Wrapper Scan Protocol**
  - Captures data at cell input
  - Shifts data from scan input (SI) to scan output (SO)
  - Updates shift stage data to update stage

# TAM Connection Example-- Core with Parallel Internal Scan



- Core internal scan path & Wrapper Cell Register are connected in parallel to TAM by a Core Test instruction
- Many other TAM connections and configurations are possible!

# **SOC Test Methodology**

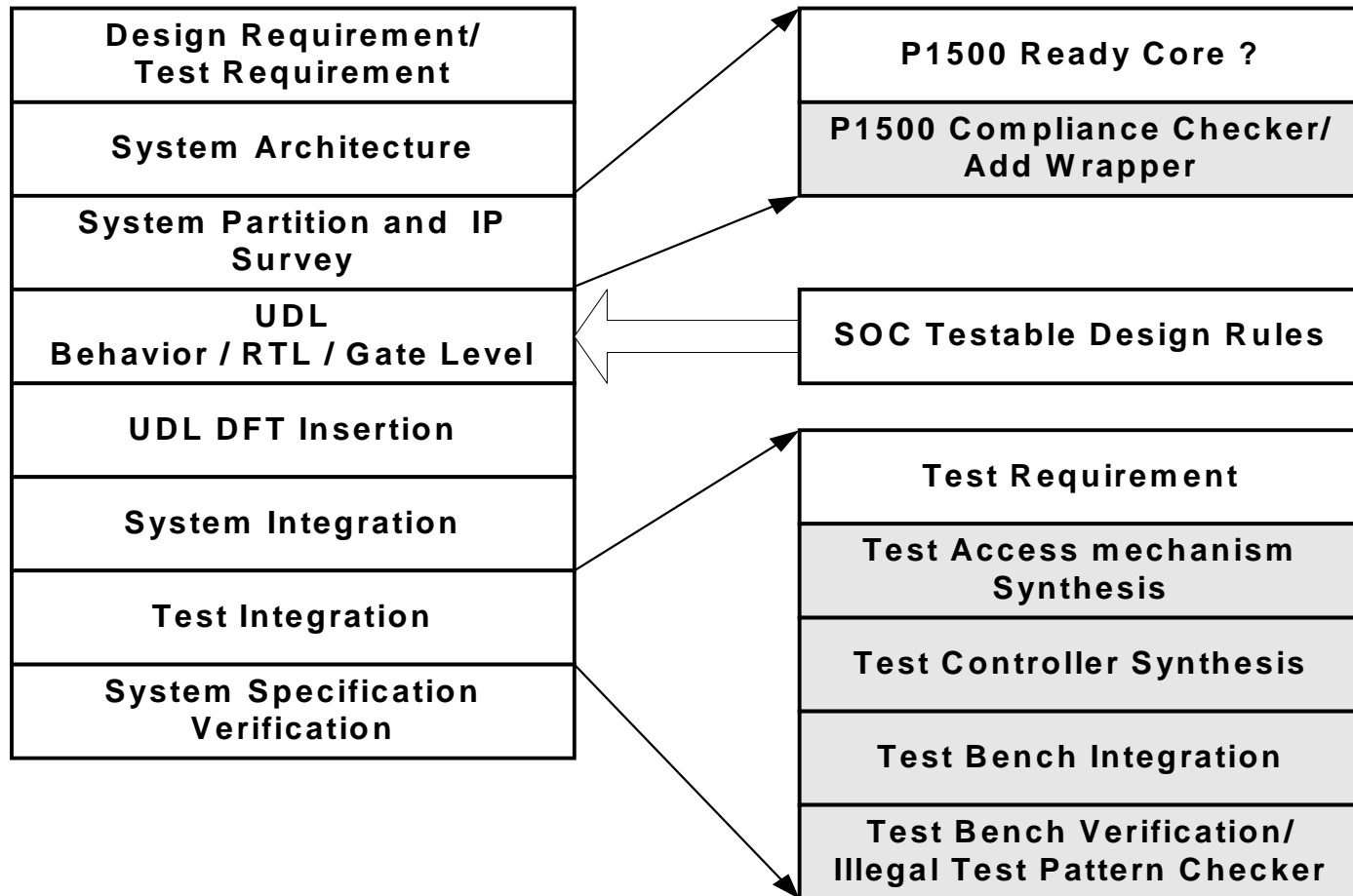
- **Study functions and architectures in each module of a general SOC**
- **Design each module**
- **Apply proper testing methods to each module**
- **Add wrapper to each core (module)**
- **Integrate the IP testing using a P-1500 like structures**



# Development of Testable SOC

- **Testing for digital components**
- **Testing for analog components**
- **Testing for memory components**
- **Wrapper for each core**
- **Define Test Access Mechanism**
- **Test integration**
- **Testable design flow**

# SOC Testable Design Flow



# Conclusions

- **SOC testing is a must**
- **Standard not defined yet**
- **Even standard is defined, many details need to be implemented**
- **Component testing needs to consider test reuse**
- **Automation of wrapper generation & system chip interface must be done**
- **Tools for linking design flow**
- **Test access mechanism is to be user-defined, hence test engineer will not lost job**
- **Mixed-mode testing in SOC is urgent**