IEEE P1687 (IJTAG) Draft Standard for Access and Control of Instrumentation Embedded Within a Semiconductor Device

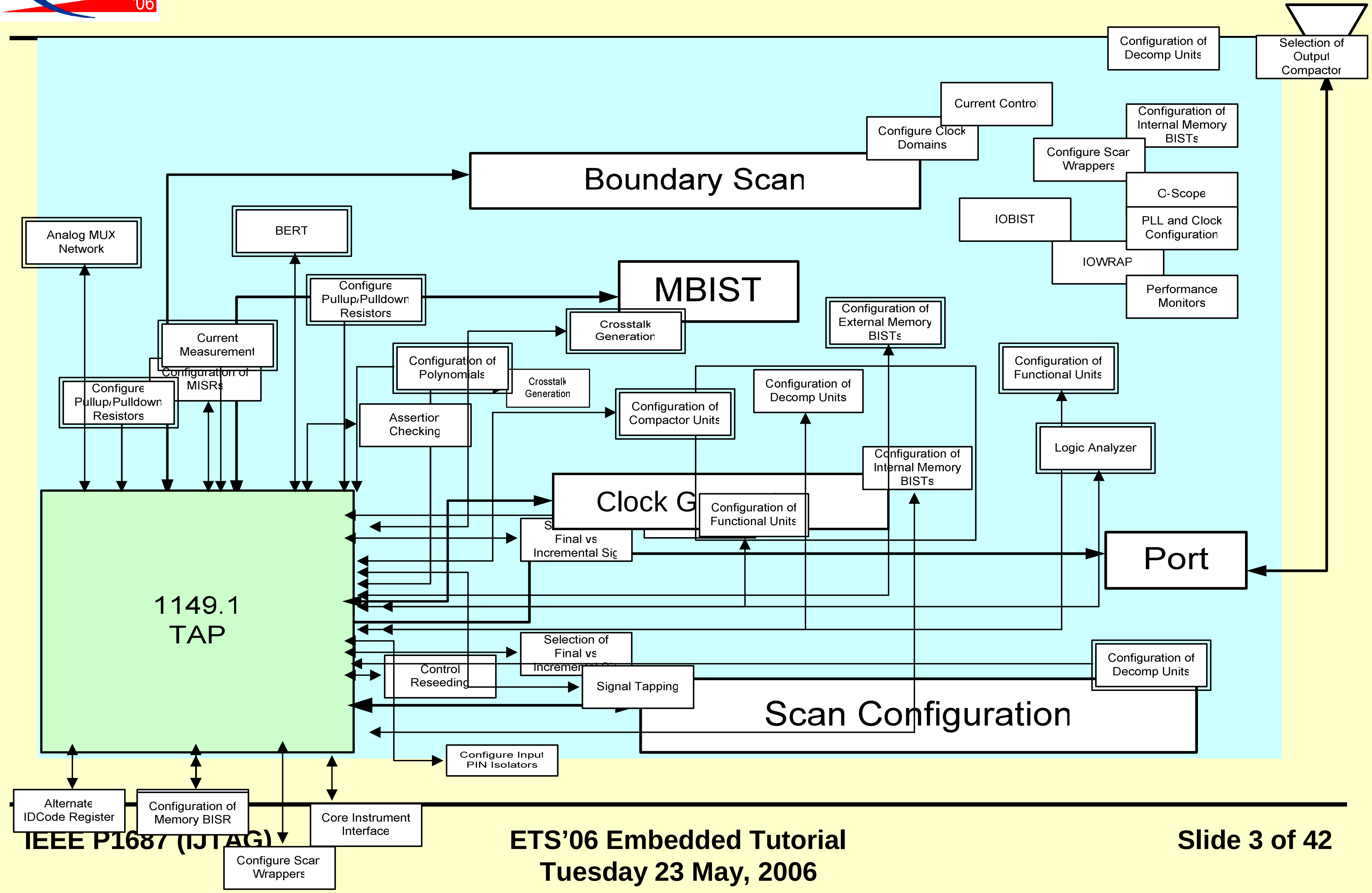
Presentation created by the IEEE P1687 Working Group.

Presented by Bill Eklow (Cisco Systems) & Ben Bennetts (Bennetts Associates)

Outline

- Background
- Describing Embedded Instrument Features
- Standardizing Access to Embedded Instruments
- High Speed Interfaces
- Other Work
- What’s next

The Evolution of Embedded Instruments



Configuration of Decomp Units
Selection of Output Compactor
Current Control
Configure Clock Domains
Configuration of Internal Memory BISTs
Configure Scan Wrappers
Boundary Scan
C-Scope
IOBIST
PLL and Clock Configuration
IOWRAP
Performance Monitors
Analog MUX Network
BERT
Configure Pullup/Pulldown Resistors
MBIST
Crosstalk Generation
Configuration of External Memory BISTs
Current Measurement
Configuration of MISRs
Configuration of Polynomials
Crosstalk Generation
Configuration of Functional Units
Configure Pullup/Pulldown Resistors
Configuration of Decomp Units
Assertion Checking
Configuration of Compactor Units
Logic Analyzer
Configuration of Internal Memory BISTs
Clock G
Configuration of Functional Units
Final vs Incremental Sig
Port
1149.1 TAP
Selection of Final vs Incremental Sig
Control Reseeding
Signal Tapping
Configuration of Decomp Units
Scan Configuration
Configure Input PIN Isolators
Alternate IDCode Register
Configuration of Memory BISR
Core Instrument Interface
Configure Scan Wrappers

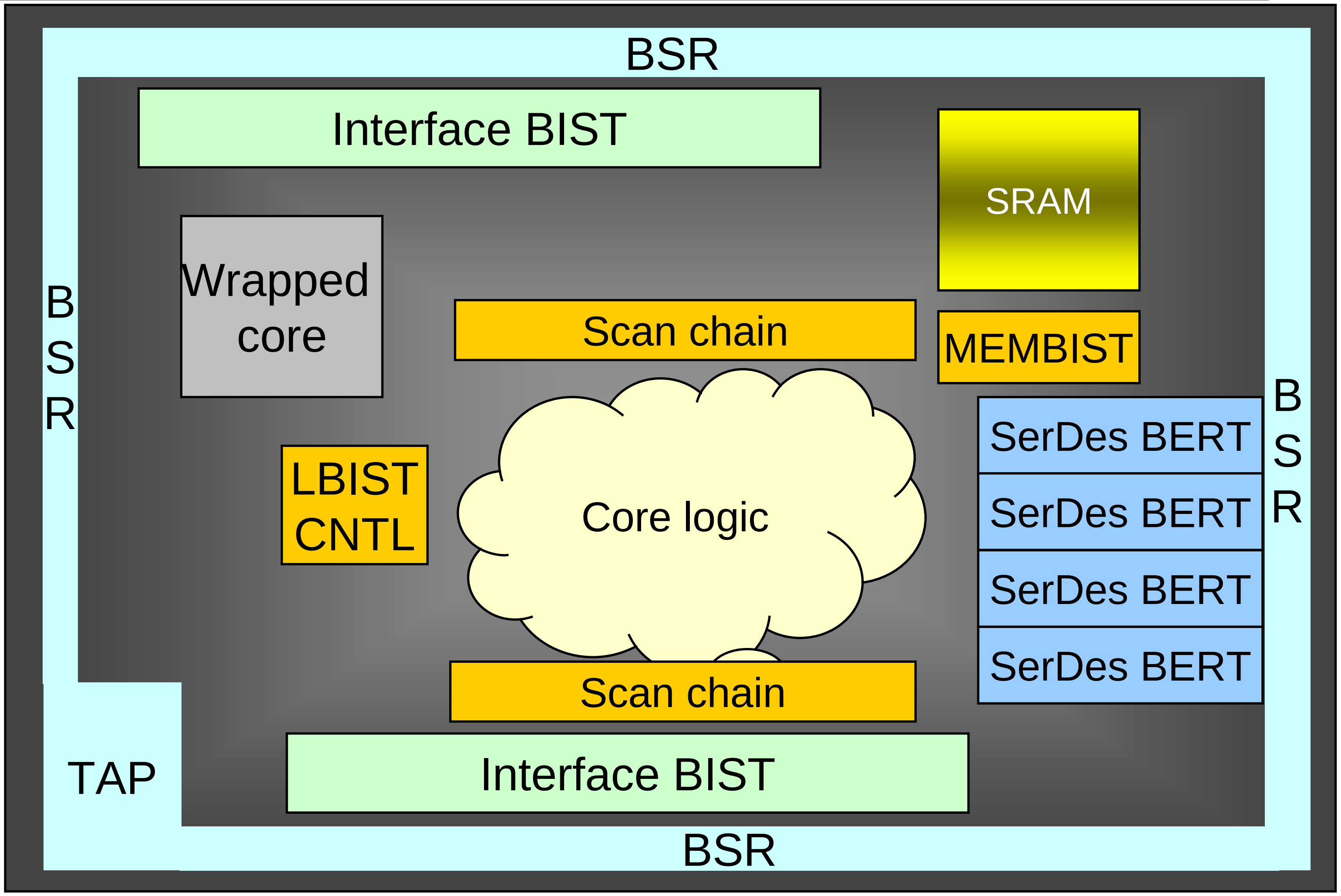
Embedded Features

1500 Access
Clock control
Power control/Measurement
Voltage Measurement
Temperature Measurement
I/O configuration
In-Circuit Emulation
Functional Configuration
Internal Memory BIST
External Memory/TCAM BIST
Logic BIST
IO - PRBS
IO - BERT
IO - Jitter
IO – SSO/Crosstalk
IO – Parametric Adjustment

Logic Analyzer
O-Scope
External Trigger selection
Performance Monitors
PLL and Clock configuration
Power Control
Scan Dump Control
X-Mask control
IBIST
Inter-domain Synchronizers
Internal Test Management
Waveform Generation/Analysis
Packet Generation
Internal Counters/Status Registers
Chip/Die ID
Analog Muxing

TAP Access to Chip Test Features

- ❑ Power management
- ❑ Clock control
- ❑ Chip configuration
- ❑ Memory test
- ❑ Scan test
- ❑ Logic BIST
- ❑ Debug/diagnosis
- ❑ PLL control
- ❑ Reduced pin count test
- ❑ Fault insertion
- ❑ Embedded instruments



P1687 Background

- Began as discussion between Agilent and Cisco
- First meeting at ITC04 BTTAC meeting
- Working group formed shortly after VTS04
  - 9 core members
  - Over 70 “extended” WG members
- PAR approved on March 16

IEEE P1687 PAR

<u>Draft Standard for Access and Control of Instrumentation Embedded Within a Semiconductor Device:</u>

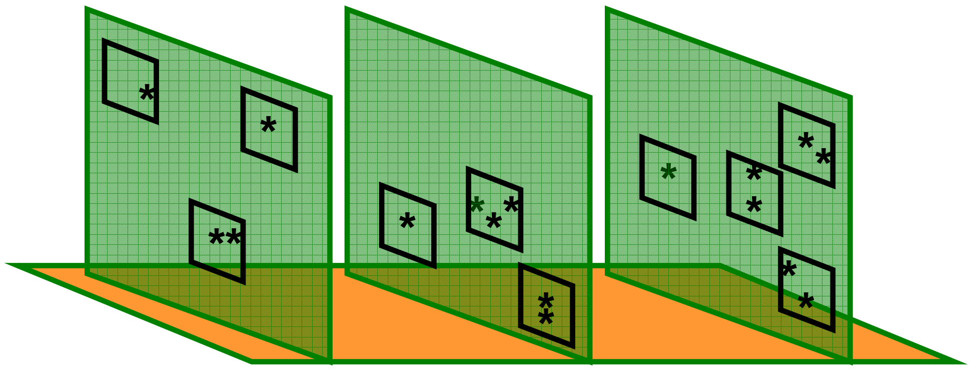
Scope: "This Standard will develop a methodology for access to embedded test and debug features, (but not the features themselves) via the IEEE 1149.1 Test Access Port (TAP) and additional signals that may be required. The elements of the methodology include a description language for the characteristics of the features and for communication with the features, and requirements for interfacing to the features"

Motivation for P1687

- Board/System:
  - Facilitate test (and debug) development
  - Interoperability of tests across multiple vendors
  - Reuse of test across multiple test processes
- Chip/IP providers:
  - Test features become product features
  - Better ATE to System correlation
- Chip ATE:
  - Facilitate test (and debug) development
  - Promote Low Cost chip ATE

Example: Creating a Board/System Test

- Complex board/system
  - Multiple SOC ASICs
  - Microprocessors
  - Embedded and standalone memories
  - Programmable logic devices
- Complex features
  - High-speed I/O
  - Backplane connections
  - Reconfigurability



Goal: Re-use Component Test Features ...

... at board and system level:

- ❑Re-run ASIC embedded memory tests
- ❑Re-run ASIC logic BIST
- ❑Run ASIC-based external memory tests
- ❑Run chip-to-chip HSIO PRBS tests
- ❑Monitor internal signal waveforms
- ❑Capture internal chip state
- ❑Use chip test features to test board

❑... and even at chip level

Daunting Task: Assembling ASIC Info

- ❑ BSDL (Boundary Scan Description Language)
- ❑ Initialization sequence(s) and clock control
- ❑ Logic BIST and external MBIST recipes
- ❑ For each embedded memory
  - Setup, launch, checking procedures
  - Diagnostic routines
- ❑ List of extra test features and access methods

More Information to Collect: Board/Sys

- For each high-speed link
  - Method to setup, launch, and check BER
  - Ability to apply different crosstalk, jitter, noise, data content conditions
- For each parallel bus
  - Method to setup, launch, and check SI
  - Patterns for crosstalk, glitches, etc.
- For each backplane configuration
  - …

Accessing Test Features is Painful

- Multiple ASIC vendors
- Multiple memory vendors
- Multiple test methodologies
- Multiple ATE platforms
- Multiple test languages
- Bottom line from example:
  - Chip test re-use at board/system: tough!

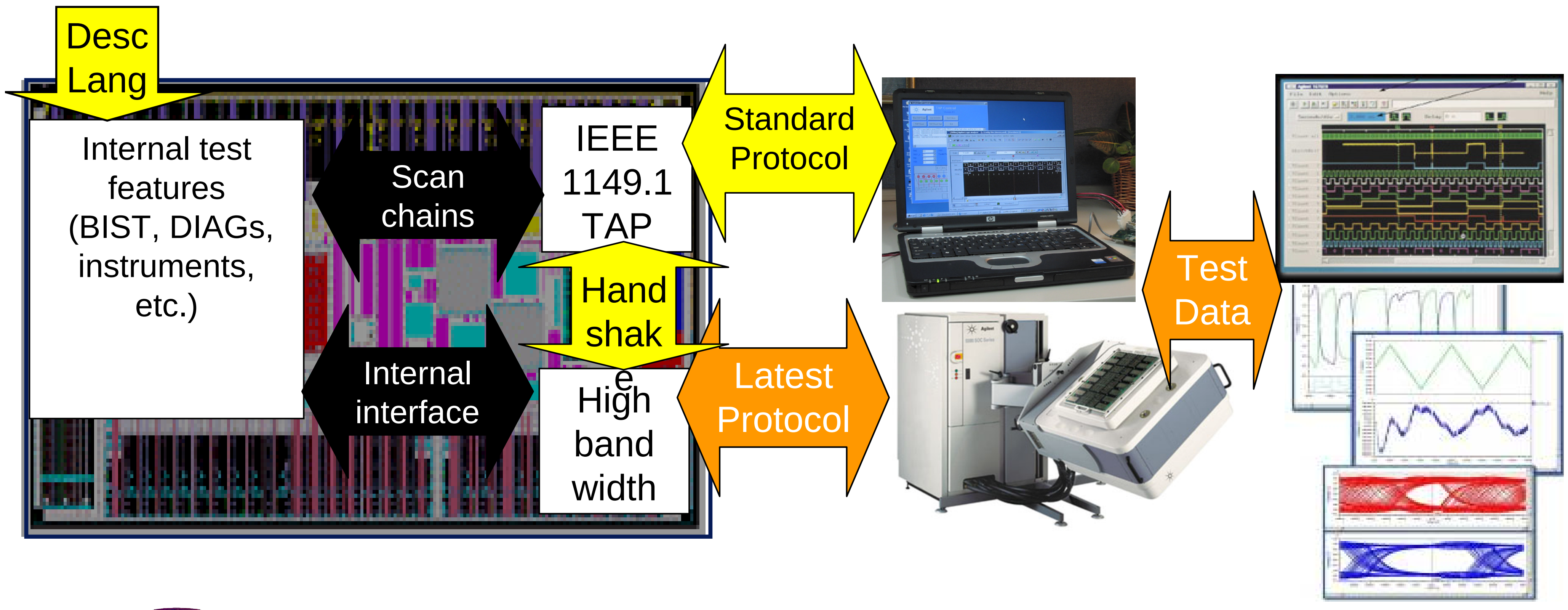
P1687 Problem Statement

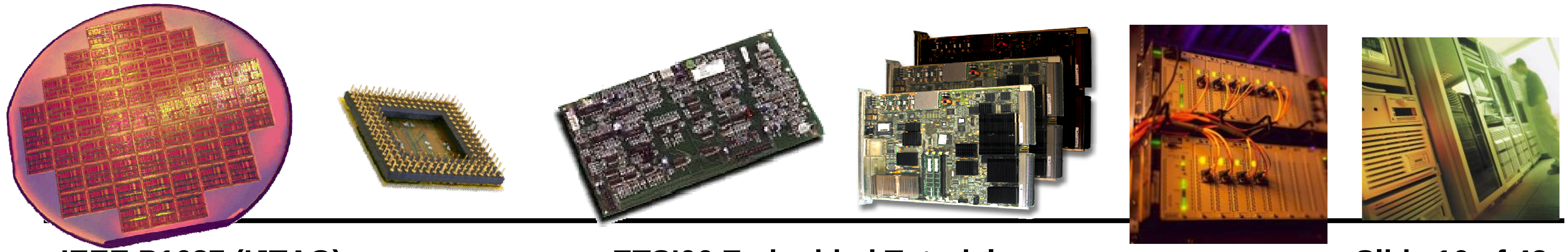
- There is currently no standard method to describe or interact with chip instrumentation IP
- It is currently very difficult to exploit the wealth of chip-level DFT/DFD features at the board & system levels
- There is a growing supply of test/debug IP and a growing need to use it at higher levels….

Key Focuses of P1687

- Documentation:
  - Identify “accessible” embedded instruments
  - Specify “characteristics” of the instrument
- Access Protocols:
  - Describe how to communicate with an instrument
  - Facilitate reuse through portability
- “Enhanced”, Secondary Access/Interface:
  - Service instruments not easily handled by boundary-scan (i.e. use high bandwidth I/O)
  - Simplify hierarchical test architectures

P1687: TAP-based Access to Test Features





Documentation - Current Approach

- ❑ Communication between provider and consumer is ad-hoc
- ❑ No mechanism to specify which features are included in the chip
  - In most cases available test instruments are not used due to lack of knowledge
- ❑ Details often included in long specs (no common format)
  - Many times specs are communicated "word of mouth"

Survey of BIST Documentation

- Supplier MBIST and IOBIST
  - Brief description of logic
  - Boundary scan tester macros
- Cisco Internal Logic BIST
  - No HW documentation
  - Verilog test bench
- Supplier PRBS
  - No documentation (application engineer)

Survey of BIST Insertion

- Cisco Internal Scan Dump
  - Macros provided by ASIC DFT group
- Cisco External Memory BIST
  - 50+ page specification
  - Full HW and access protocol definition

Documentation – Intent

- ❑ End user can easily identify and program embedded instruments
- ❑ Facilitate automated tests based on machine readable descriptions
- ❑ Minimize “Time to Understanding” and “Time to BringUp”

P1687 HW Documentation Requirements

- Describe the ARCHITECTURE of internal instruments i.e. how to use, but not what they do.
- Provides inventory of instrument content on a chip
- Necessary to identify number and location of instruments
- Enough information for a programmer to figure out how to perform low level functions
  - Instrument name, type and instance
  - Register definition (location and length)
  - Bit definitions
  - Data/Instruction formats
  - Internal/External dependencies

P1687 BSDL Attribute Proposals

- Instrument definition

```
attribute INSTRUMENT_DEF of <device name>:
        entity is
        "Core1          IP1," &
        "Core2          IP1";
```

- Chain definition

```
attribute CHAIN_DEF of <device name> entity is:
// first register in chain is R1 of Core 1 and it is 4 bits
          "Chain1 (Core1.R1,4)," &
          "Chain1 (Core1.R2,4)," &
// first two elements of Chain2 are R1 and R2 of Core2
          "Chain2 (Core2.R1,4  Core2.R2,4)";
```

- Register access

```
attribute REGISTER_ACCESS of <device name> entity is:
        "Chain1 (MEMTST1)," &
        "Chain2 (MEMTST2)";
```

Common Access Protocol – Current Approach

- Today's "common" I/Fs for embedded instruments:
  - TAP-based
  - I2C
  - Custom CPU interface
  - Internal core-based
  - Custom ATE / interface

$\left.\vphantom{\begin{matrix}a\\b\\c\\d\\e\end{matrix}}\right\}$ *Each with its own language(s)!*

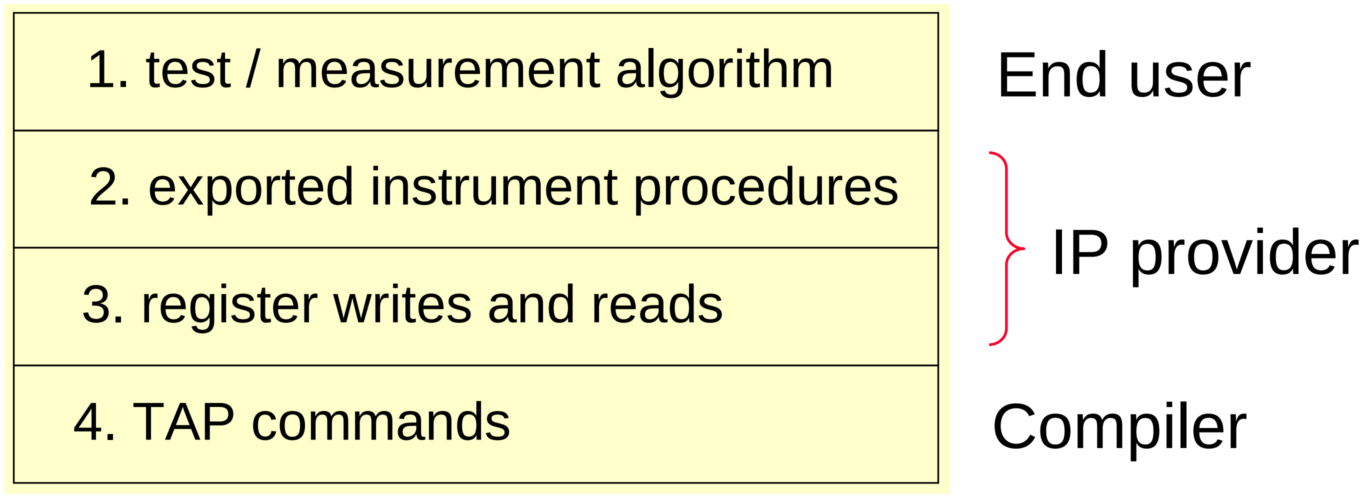
Common Access Protocol - Intent

- Facilitate reusable code across boundary-scan platforms and different test processes
- Simplify program development by building on several lower level procedures e.g. IR-Scan, DR-Scan, etc
- API vs. Language?

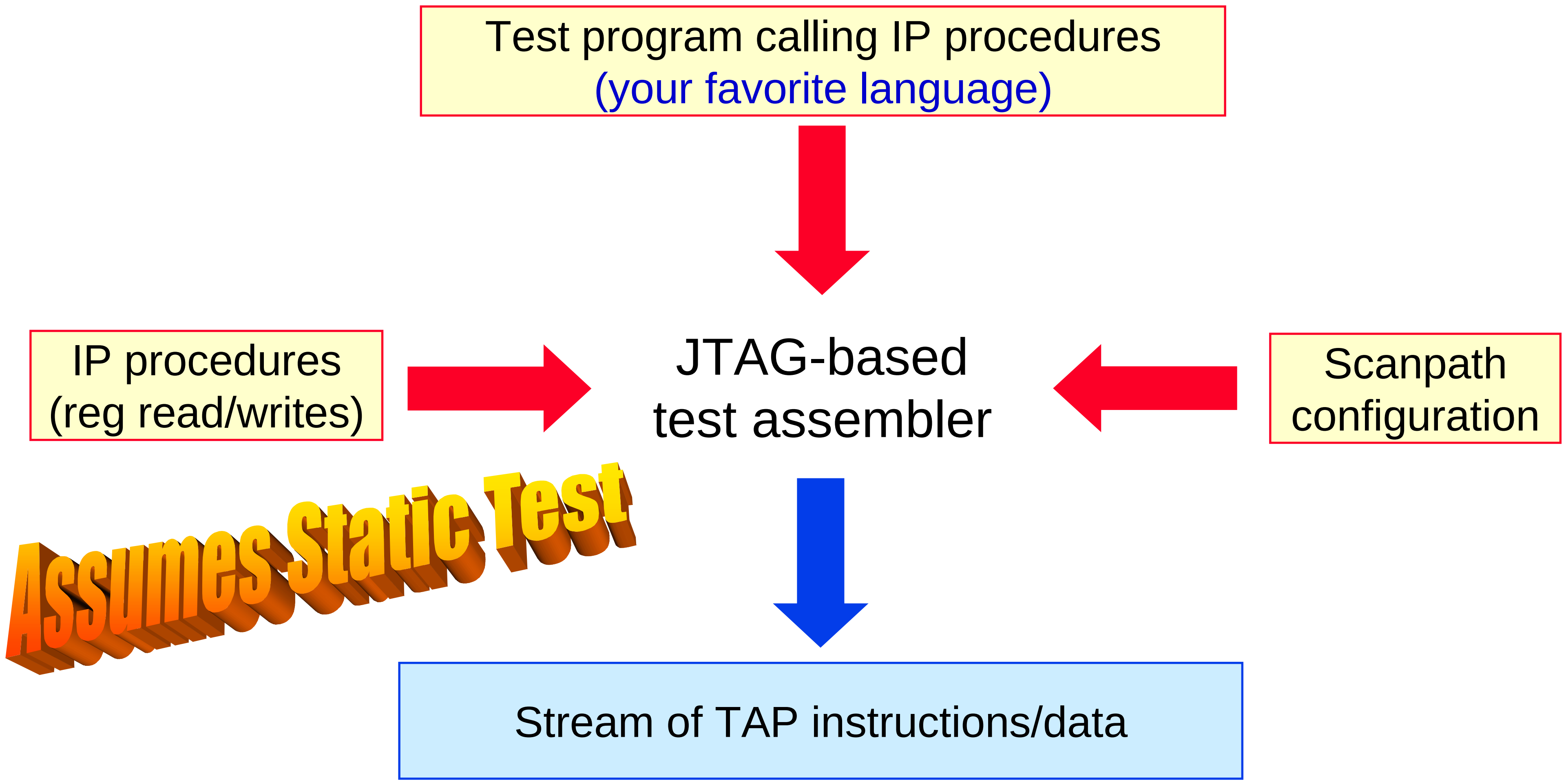
IEEE P1687 API Model

- P1687 procedures can be thought of as an API:
  - **Can be called from a variety of higher-level environments**
  - **Delivered as a package by IP provider**
  - **Expose only those features that IP provider chooses**
  - **Hide low-level details from user**

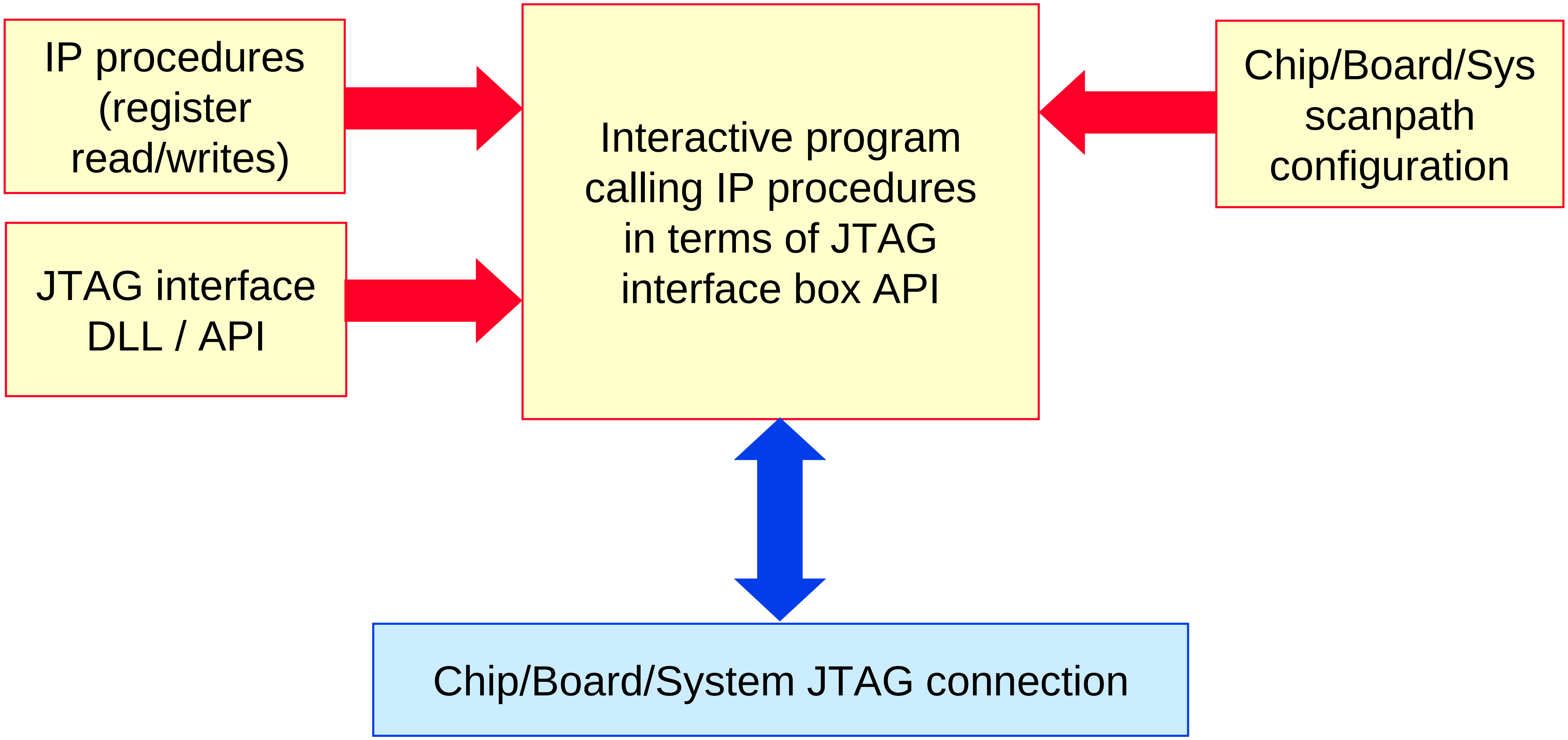
- Layers:


1. test / measurement algorithm
2. exported instrument procedures
3. register writes and reads
4. TAP commands
End user
IP provider
Compiler

P1687 Static Test Assembly Flow



P1687 Interactive Test/Debug Env


IP procedures (register read/writes)
JTAG interface DLL / API
Interactive program calling IP procedures in terms of JTAG interface box API
Chip/Board/Sys scanpath configuration
Chip/Board/System JTAG connection

Memory BIST Procedure

- Function declaration
  - **e.g. : int run_mbist_ram256x32sp(int repair_enable, int background);**
- Function body

```
int run_mbist_ram256x32sp(int repair_enable, int background);
    {
     start_clock(MCK);
      done_reg = 0;
      pass_reg  = 0;
      repair_enable_reg = repair_enable;
      background_reg = background;
      start_BIST = 1;
    }
```

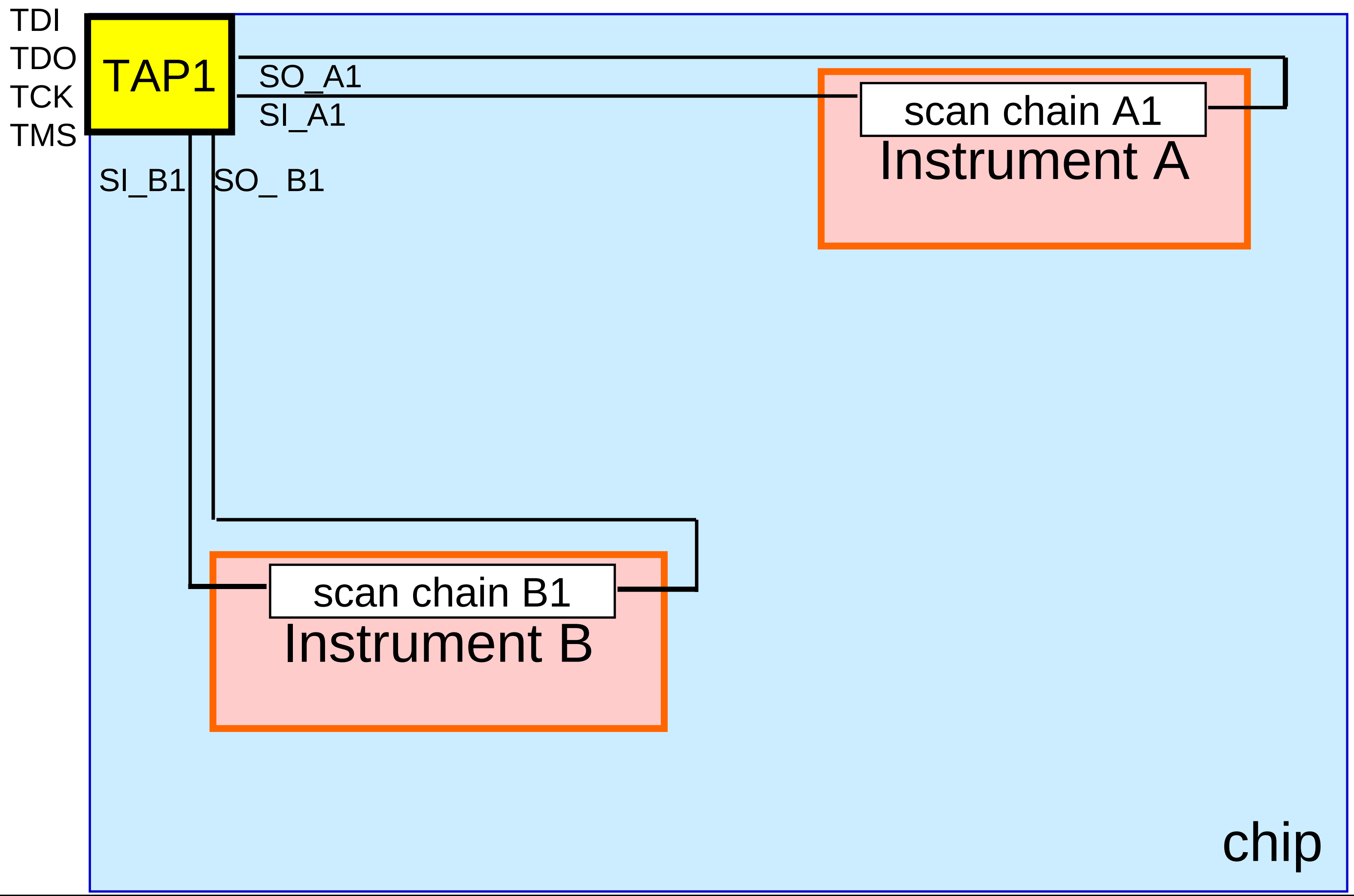
Secondary Interfaces/HUB - Intent

- ❑ Allow hand-off to an interface with higher bandwidth than the 1149.1 TAP for data-intensive operations
- ❑ Allow interoperation between individual instruments that may require asynchronous signaling
- ❑ Allow an avenue for support of legacy (non-TAP) instrument interfaces

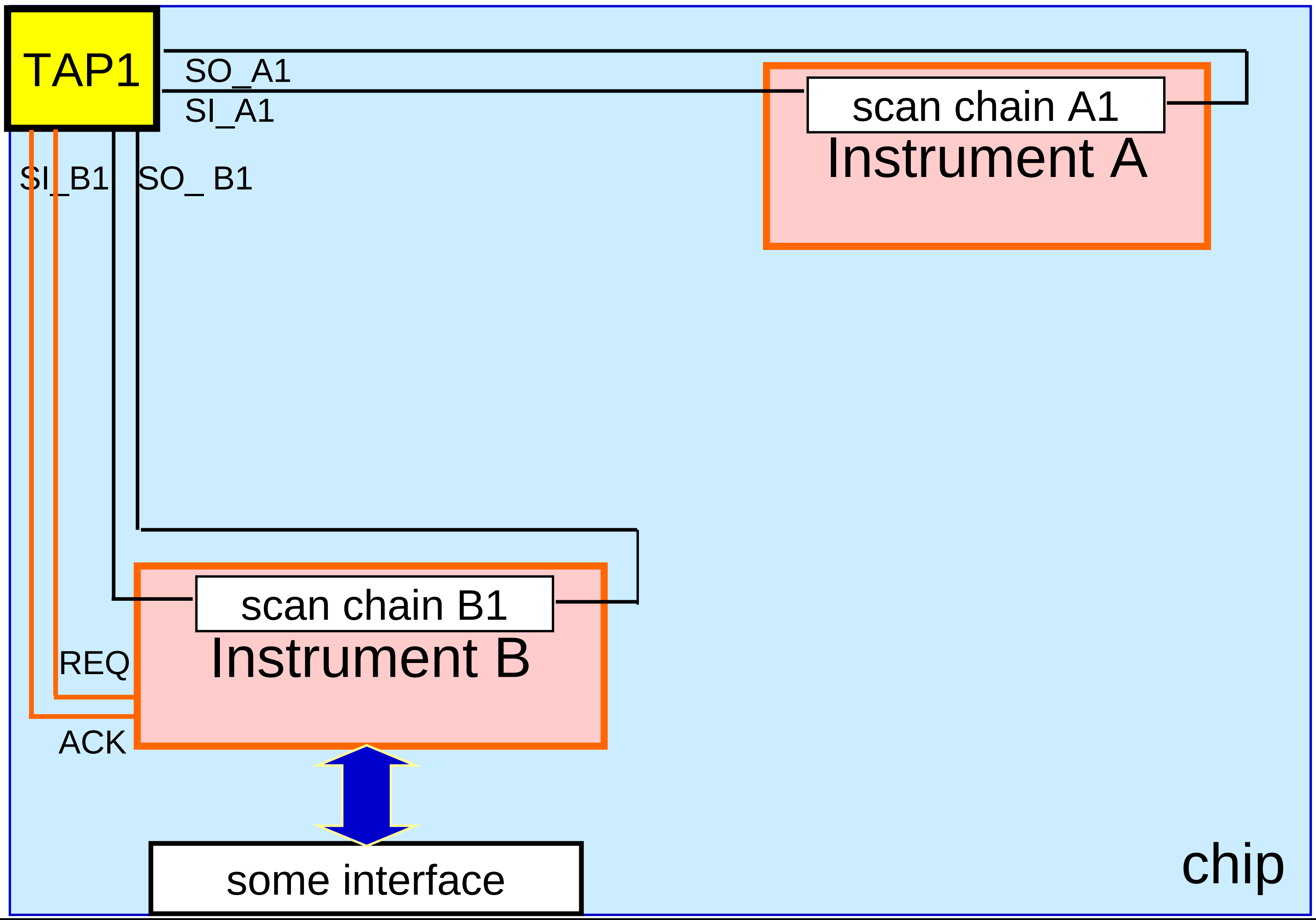
Evolving Interface Requirements

- ❑ Phase 1: simple instrument-to-TAP interface
  - Need: spec for standard interface to instruments
  - Solution: scan-based: TAP Test Data Registers (TDRs)
- ❑ Phase 2: higher bandwidth I/O
  - Need: hand off data transfer to another interface with higher bandwidth than TAP
  - Solution: mux controls to configure I/Os
- ❑ Phase 3: instrument intercommunication
  - Need: instrument-to-{instrument/ATE} comm
  - Solution: hierarchical “HUB” with async signals

Phase 1: TDR-Based Access

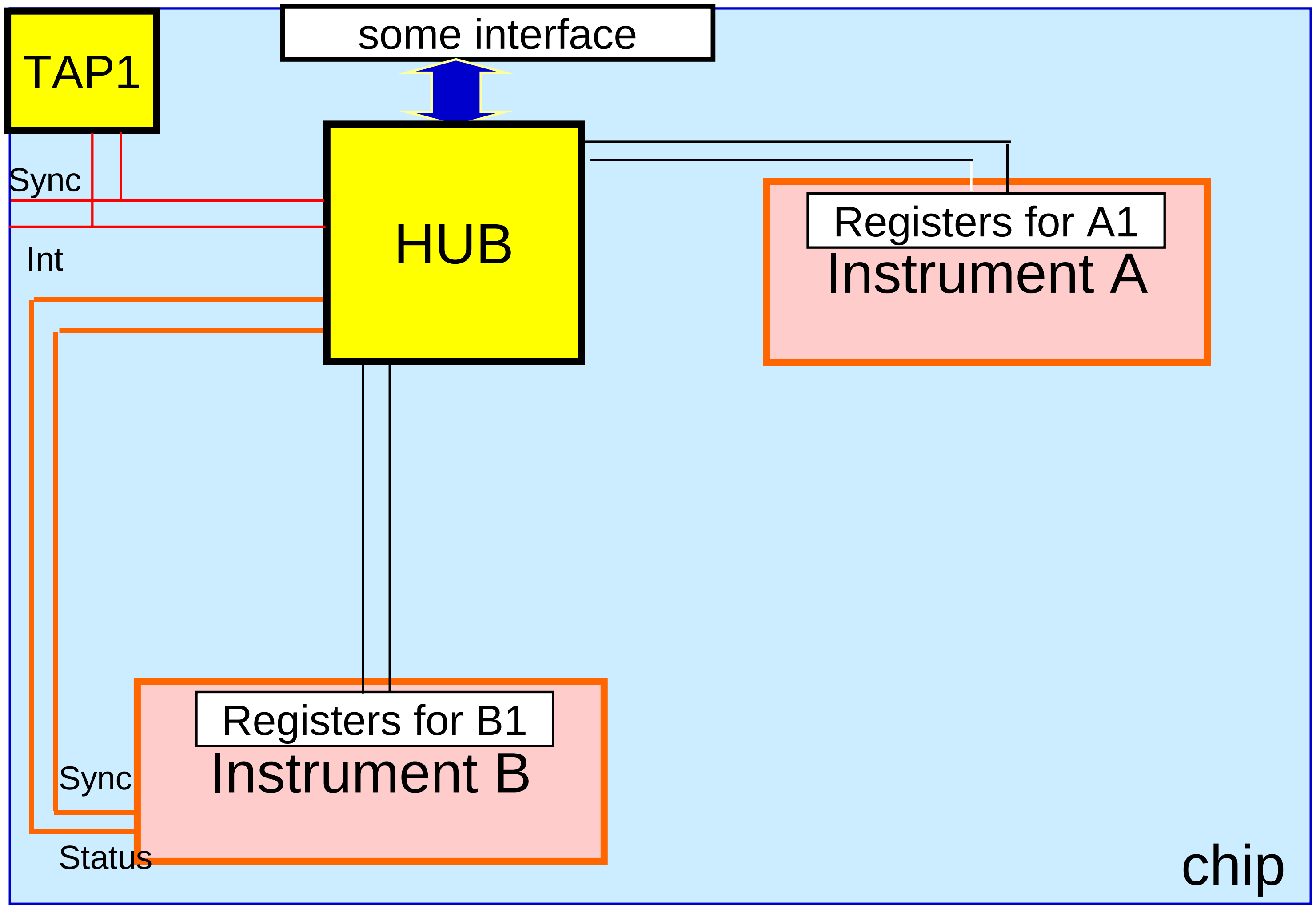


Phase 2: TDR + I/O + Polling signals

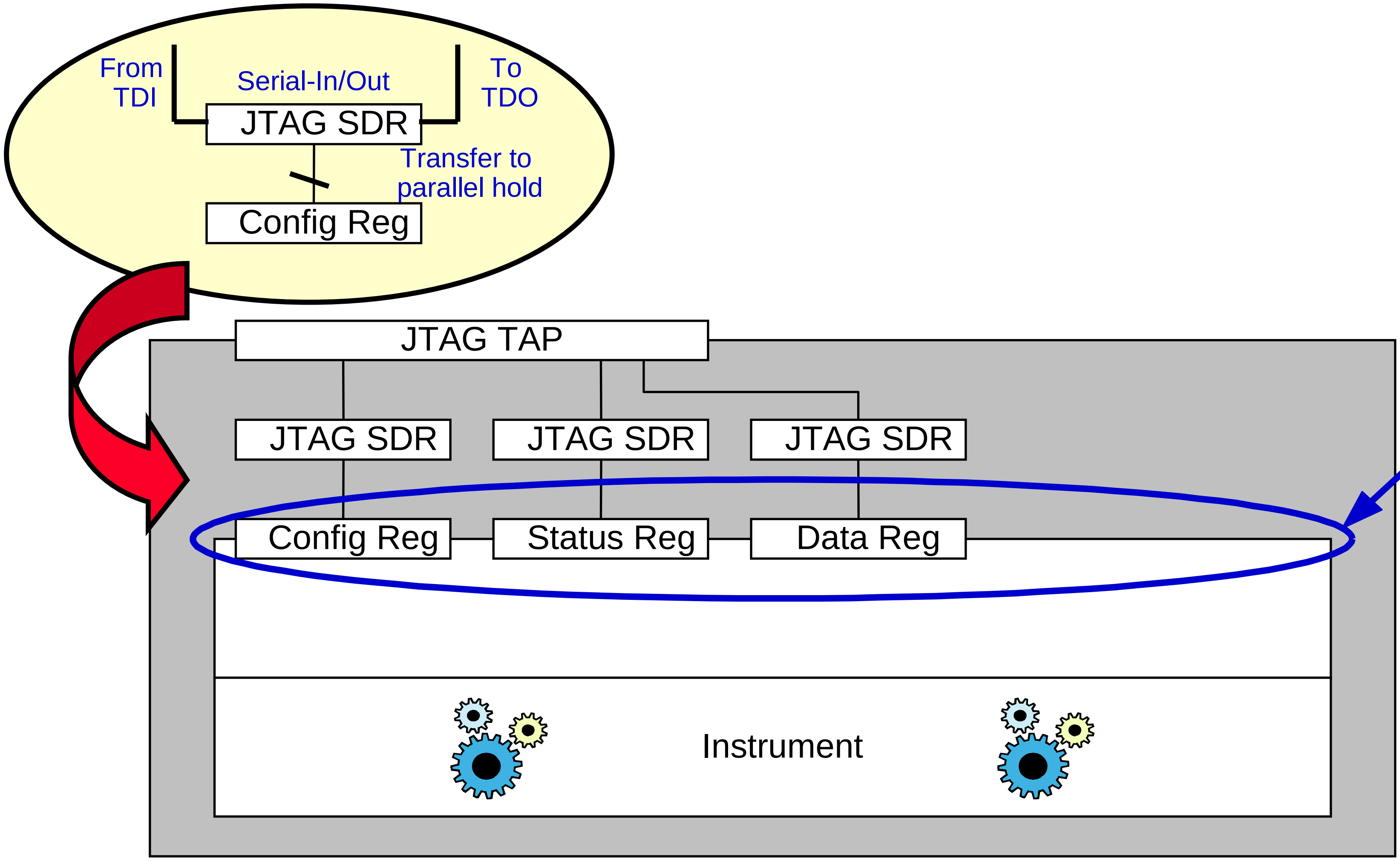


Note:
REQ = start
ACK = done

Phase 3: HUB Approach

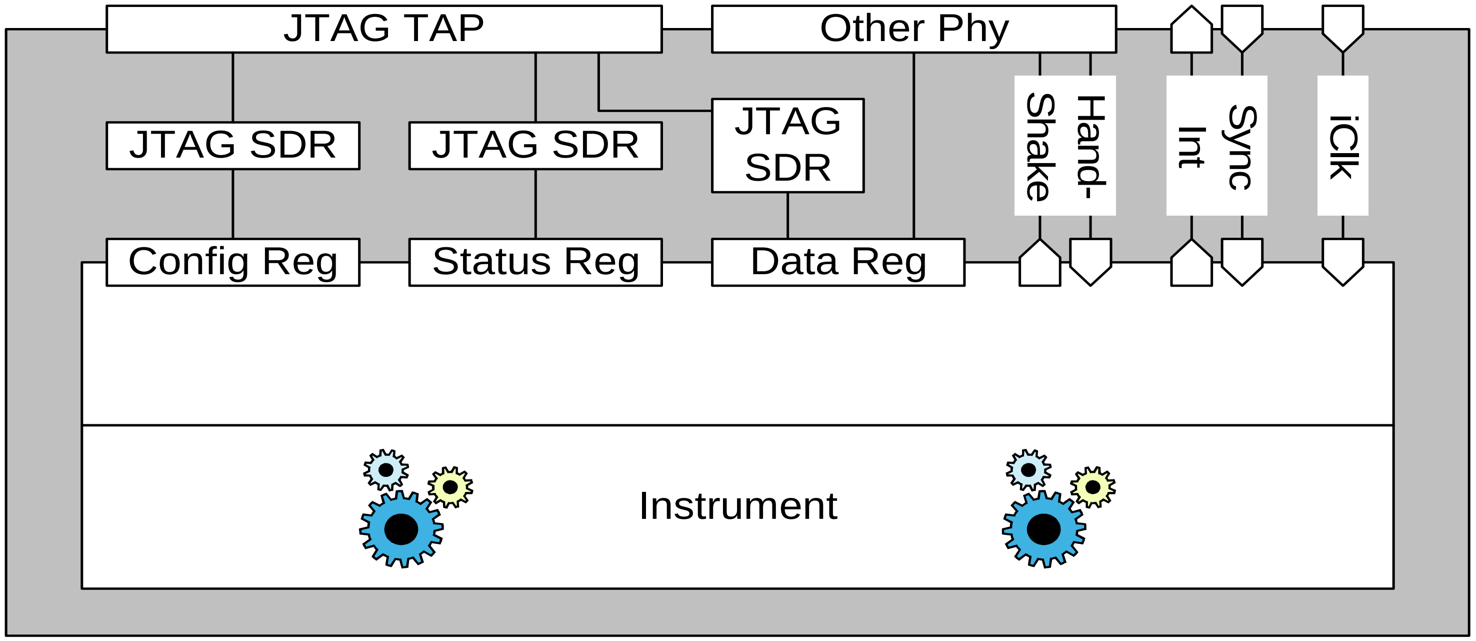


Hub-less P1687 Interface (Phase 1)

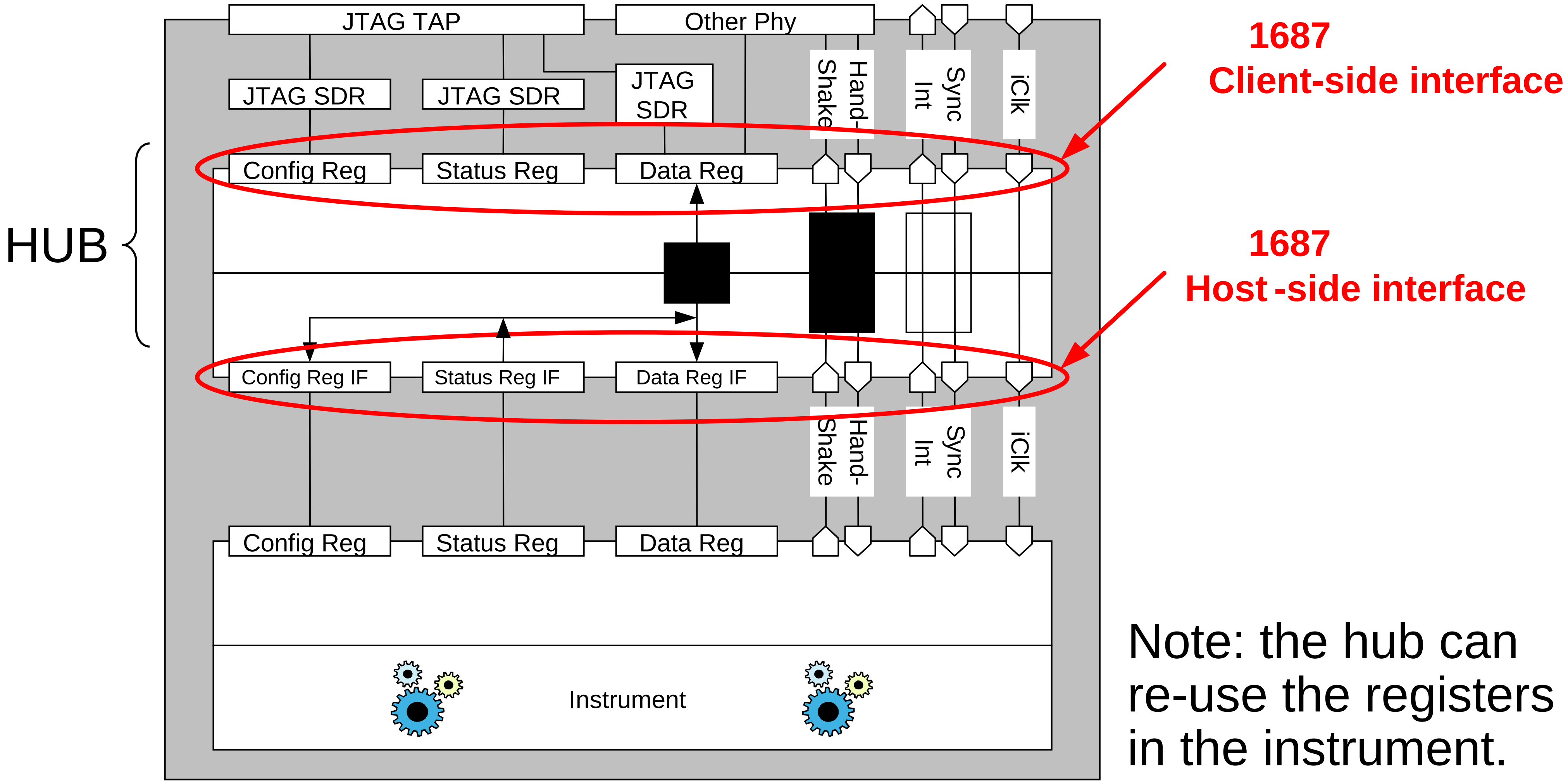


Simplest 1687 Interface

Hub-less P1687 IF w/ Phy + Async Sigs (Phase 2)

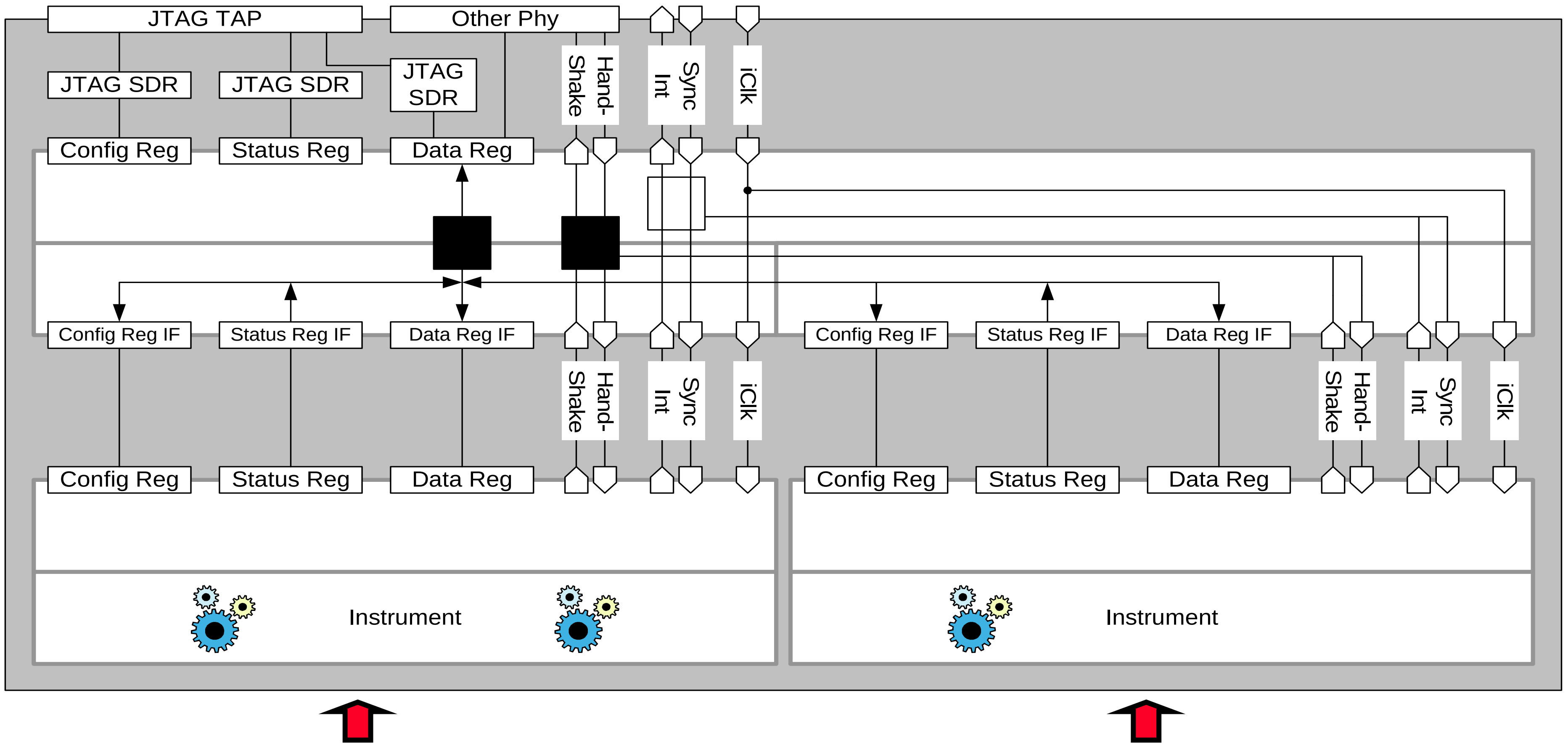


Hub: One Instrument, One Phy



Note: the hub can re-use the registers in the instrument.

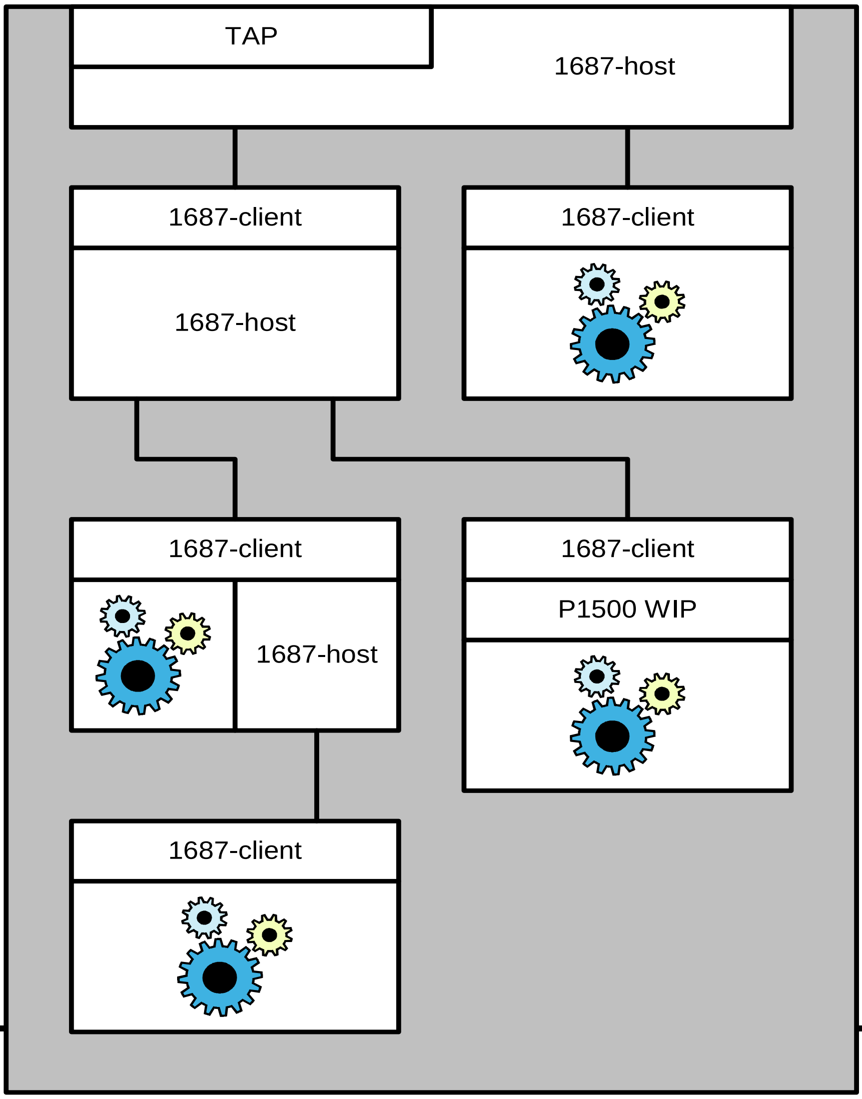
Hub with Two Instruments (Phase 3)



Could be synchronous

Could be asynchronous

P1687 Interface Hierarchy



Hub Summary & Issues

- Goal: Make the HUB just simple enough
  - Enable TAP-based interaction with instruments
  - Enable high-bandwidth data transfer (optional)
  - Enable instrument interaction (optional)
  - Enable cascading hierarchically (optional)
- Status: Closing in on an architecture

Open Issues

- ❑ Exact architecture of the Hub
  - Options: Xilinx ICON; Altera System-Level Debug (SLD); LV WrapperTAP
  - Mandatory connection to the TAP?
- ❑ BSDL for architectural description: syntax, specs
- ❑ Procedural language choice(s):
  - Simple stream of register reads/writes?
  - Full-featured programming language?
- ❑ Compliance checking
- ❑ HELP!!
  - Staff subcommittees;
  - provide **real world** examples

ETS '06

IEEE P1687 Summary

- ❑ IJTAG fills a very real need
- ❑ IJTAG scope is focused to succeed
  - Standard description of internal features
  - Standard protocol for access
  - High bandwidth interface mechanism(s)
- ❑ IJTAG gaining momentum
- ❑ IJTAG needs your input!
- ❑ To get involved/learn more/register your interest, contact Ken Posse (IJTAG Chairman) at **kepos@comcast.net**

IJTAG

Any Questions?