# IEEE 1687 IJTAG
# HW Proposal

# 1687 Proposed Hardware Architecture

This document contains the key components of the proposed 1687 Hardware Architecture that were recently adopted with a working group acceptance vote on 5 May, 2007. These components are the collection of all of the hardware elements and concepts that the Voting Committee tentatively accepted as the Strawman Architecture to consider at the start of the language effort.

The vote did not fix the HW Proposal so that it is unchangeable, but just accepted it at this point in time – and it is fully expected that items in the HW Proposal will be revisited, modified, and changed as the needs and limitations of the language and automation support evolve and develop.

Note: it is assumed that the users of this IEEE 1687 summary document are knowledgeable on IEEE 1149.1-2001 and IEEE 1500-2005.

## Quick Terms, Acronyms, Abbreviations, & Concepts Introduction

**There are many terms, acronyms, and new concepts that will be introduced in the course of the Hardware Proposal Summary – the most common and prevalent ones will be listed and defined here. A full Glossary is included after the main body of the Hardware Proposal Summary document.**

**Note: the following style formats are used in this document:**

IEEE 1149.1 instructions are written upper case and bold e.g. **PRELOAD**

IEEE 1149.1 state-machine states are written mixed case and italicized e.g. *Shift-DR*

Signal names are written mixed case and bold e.g. **Select-WIR**

Rule directives are written in lower case and bold e.g. **requires**

- **IEEE 1149.1-2001:** the board-test standard that defines an on-chip test-access-port (TAP) of 4, optionally 5, pins and associated signals and mandatory on-chip logic (Instruction Register, 16-stateTAP Controller, Bypass Register and Boundary-Scan Register) that is used to verify device connectivity in the board-environment. The standard is sometimes synonymously referred to as JTAG.

- **JTAG**: the acronym for the Joint Test Action Group - the group identity used by the original creators of the 1149.1 standard. JTAG is often used to mean the 1149.1 standard itself rather than the group of originators.

- **1149.1-Zone**: the compliant board-test portion of the device that is the implementation of the 1149.1 Standard and includes all of the required portions, and may contain some of the optional portions described in the 1149.1 Standard – items such as the 4 or 5 pin Test-Access-Port (TAP), 16-state TAP Controller, Boundary-Scan Register (BSR), the Bypass Register, the 1149.1 Instruction Register, and the IDcode Register.

- **1149.1-Overlap-Zone**: the compliant board-test portion of the device that is the implementation of the 1149.1 Standard when it also includes a 1687 Gateway and the instruction that selects and configures the Gateway.

- **1149.1-IR**: the compliant and defined 1149.1 Instruction Register that supports the defined required and optional Public Instructions – for example, **EXTEST**, **SAMPLE**, **PRELOAD**, **HIGHZ**, **IDCODE**, and **CLAMP**.

- **1149.1-SM**: the compliant 1149.1 16-state State Machine that supports a *Test-Logic-Reset* state; a *Run-Test-Idle* state; a *Select-DR Scan* set of states that allow DR-Scans; and a *Select-IR Scan* set of states that allow IR-Scans.

- **1149.1 Compatible**: any instrument or element that operates using only the signals and sequences available from a fully-compliant 1149.1 TAP and TAP Controller.

- **IEEE 1500-2005:** core-test standard that defines core-test-wrappers and the core test-access-mechanism (TAM).

- **TAM**: Test Access Mechanism – the connectivity and protocol structure used to access an instrument. For the IEEE 1500, a defined TAM exists.

- **WIR**: Wrapper Instruction Register – the local instruction register associated with the IEEE 1500 TAM.

- **IEEE P1687:** the on-chip instrument access and control standard that defines the connectivity and interfaces of embedded instruments other than those associated with 1149.1 that are used for board-test.

- **Hierarchical Element**: an element is a standalone Test-Data-Register (TDR) or TDR-Bit whose only purpose is to provide a hierarchical connection – there is no other functionality that would classify it as an instrument.

- **Instrument**: any logic structure within a device with a purpose other than just providing 1687 connectivity that is accessed by the 1687 architecture (connectivity-only is defined as an Element) – instruments can be Design-for-Test (DFT), Design-for-Debug (DFD), Design-for-Yield (DFY), Test, or Functional items.

- **Hierarchical Instrument**: an instrument that also provides a hierarchical connection by passing on, at a minimum, the Select-Instrument signal and the TDI-TDO serial scan-path which may be referred to, using the IEEE 1500 nomenclature, as the WSI-WSO serial scan-path to distinguish it from the serial scan path in the 1149.1-Zone. Basically, a Hierarchical Instrument provides access to other instruments that do not require a direct IR-Scan.

- **DR-Scan**: the use of the *Select-DR Scan* side of the 1149.1 State-Machine – this action is generally associated with accessing or using a targeted TDR.

- **Shift-DR**: a noted sub-operation of DR-Scan that represents the action of shifting data through the selected Data Register that is connected between the device's TDI-TDO when the 1149.1-SM is in the *Shift-DR* state.

- **Update-DR**: a noted sub-operation of DR-Scan that represents the action of synchronizing all change actions of the target 1149.1 Data Register or 1687 instrument that is active under the current selected instruction when the 1149.1-SM is in the *Update-DR* state at the falling-edge of **TCK**.

- **IR-Scan**: the use of the *Select-IR Scan* side of the 1149.1 State-Machine – this action is associated with accessing the 1149.1 Instruction Register.

- **Gateway**: a hierarchical enabled instrument or element whose purpose is to be the only 1687 Shift-Update construct that is allowed to be enabled by an 1149.1-IR Instruction in the 1149.1-Overlap-Zone – and once enabled the Gateway becomes an alternate local Instrument-IR that can select other 1687 instruments using only DR-Scans.

- **HIP**: Hierarchical-Interface-Port – a port on a Parent-Instrument or standalone-element used to pass the Select signal and serial scan-path connections to a Child-Instrument.

- **SIB**: Select-Instrument-Bit – a construct that supports a Hierarchical Connectivity-Scheme – it may be a standalone TDR-Bit that includes a Hierarchical-Interface-Port (HIP), or it may be an identifiable structure within a Test-Data-Register (TDR) or 1500-TAM.

## The Four Main Sections of the Hardware Proposal

**The HW Proposal covers several concepts and described architecture elements (each described more fully in later sections):**

1. The concept of the compliant 1149.1-Overlap-Zone.

2. The concept of Gateway-Elements or Gateway-Instruments as the beginning of the described 1687-Zone.

3. The concept of 4 Archetypal Instruments labeled A, B, C, and D.
    a. The defined Interface Signals and Port-Rules associated with those Instruments;
    b. The concept of Alternate-Controllers or Instrument-Interface Conversions;
    c. The concept of Bandwidth Ports.

4. The concept of 2 Connectivity-Types – Flat and Hierarchical – with:
    a. 4 Flat Connection-Schemes: Flat, Daisy-Chain, Star, Concatenate;
    b. 3 Hierarchical Connection-Schemes: Replace, Concatenate-Before, Concatenate-After.

There are items discussed at the end of the main body of this document that fall outside of the 4 main sections under discussion, and these topics should eventually be a part of the Hardware Architecture Proposal. However, these items were not detailed in the voting document as anything more than concepts, but are included as items that are identified as being important for a comprehensive architecture consideration.

## #1: The 1149.1-Overlap-Zone

**According to the 1687 Scope and Purpose, access to the 1687 portion of the architecture is through an 1149.1 Test Access Port and associated control logic (Instruction Register, State Machine, addressable Test Data Registers, etc.). To this end, there is a need to maintain an efficient and compliant 1149.1 architecture for board-test purposes and 1687 can not impose any requirements that would result in a non-compliant and non-efficient 1149.1 board-test architecture. The best way to accomplish this is to limit "what of the 1687 hardware architecture" that the compliant 1149.1 board-test portion must see and describe. Conversely, there is not a need for 1687 to replicate all items that are already defined and described by 1149.1 in order to be rated as a compliant 1687 architecture.**

**This portion of the HW Proposal puts forth the concept that 1149.1 exists; has a defined architecture and instructions; needs to remain compliant in its defined usage-space; and it is mandated that 1687 does not cause any non-compliance, and preferred that 1687 does not cause any inefficiencies, to the 1149.1 usage-space:**

***RULES:***

- It is our stated scope and purpose to **require** use of an 1149.1 compliant TAP and TAP Controller as defined by 1149.1.

- …with **no requirement** for 1149.1 to be in a Compliance-Enable mode to use or access the 1687 portion of the architecture.

- …with **no requirement** of using a device supporting 1687 apart or in isolation from other devices in a multiple 1149.1 board-test system.

- …with **no requirement** or **allowance** of placing other functions, logic or filters in front of the 1149.1 TAP (external to the device, or between the 1149.1 TAP and TAP-Controller) for 1687 purposes.

- …with **no requirement** to change any element of the publicly described portions of the 1149.1 standard: the 1149.1 TAP; the 1149.1 State Machine; the Boundary-Scan-Register (BSR); the Bypass Register; the ID-Code Register; the Instruction-Register; etc.; the Public Instructions (**EXTEST**, **INTEST**, **SAMPLE**, **PRELOAD**, **IDCODE**, **CLAMP**, **HIGHZ**, etc.).

- …with a **preference** that board-test-only instruments (that can be described in BSDL) should be included within the 1149.1-Zone and may be accessed with declared public-instructions or private-instructions.

- …with an **acknowledgement** that the Overlap-Zone of the architecture is shared by 1149.1 and 1687 and the dominant considerations of this zone is 1149.1 and so the zone will be described using BSDL as stated in the 1149.1 Standard; connection-schemes will be driven by 1149.1 requirements; and compliance-checking is in compliance with 1149.1 criteria.

- …that 1687 **formally acknowledges** that it does not want to replace or re-describe the board-test elements defined in the 1149.1 standard as part of the 1687 standard (but to reference to 1149.1-2001, for example, as the port, controller, and access to the 1687 Gateway).

- …that it is **preferred** that the inclusion of 1687 features and functions do not make for operation, engineering, or use inefficiencies of the BSDL-described board-test portion of the 1149.1 architecture.

- …that the **preferred** extent of impact to the 1149.1-Overlap-Zone, by inclusion of 1687 concepts and hardware, will be to place 1149.1-compatible instruments; and/or 1149.1-defined Test Data Registers (TDRs); and/or hierarchy-support elements within the 1149.1-Overlap-Zone that can be described by BSDL.

- …that any of these instruments that enable hierarchical-access (hierarchical-access meaning to access other instruments that do not require a direct IR-Scan) **shall** be known as Gateway elements.

- …that it is **required** to add instructions to the 1149.1 Instruction Set for the elements contained within the 1149.1-Overlap-Zone.

- …that it is **required** that the connectivity of instruments in the 1149.1-Zone be driven by 1149.1 requirements, optimizations, and compliance-checks.

- …that any 1687 instrument that cannot be described by BSDL should not be directly connected to the 1149.1-IR and **should not be allowed** in the 1149.1-Overlap-Zone (but should be moved to the 1687-Zone) – this includes instruments that may be board-test instruments but can not be described with BSDL.

- NOTE: the 1687 hardware items in the 1149.1-Overlap-Zone should be the only 1687 instruments that can react to an IR-Scan – all other 1687 instruments should be accessed, configured, and controlled using only DR-Scans (*Shift-DR* and the *Update-DR* synchronizing events).

***Examples & Clarifications:***

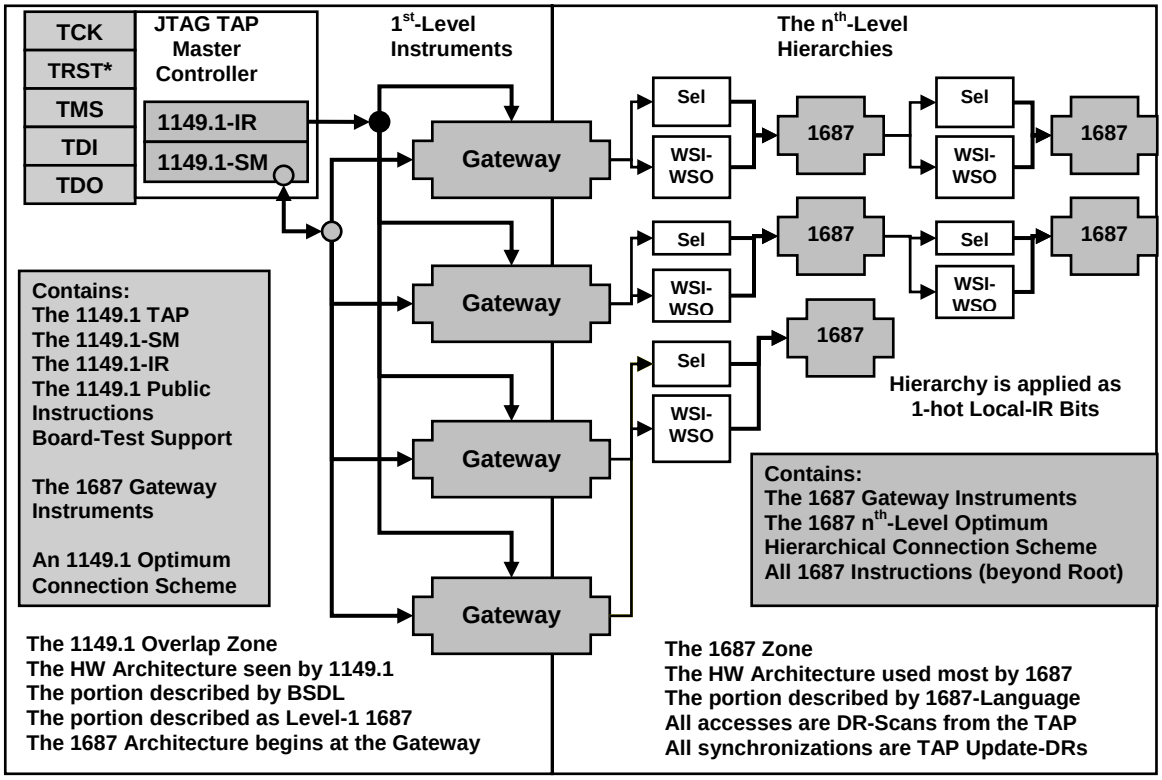


**Figure 1: Graphic View of the 1149.1-Overlap-Zone**

- Figure 1 shows a graphical example of the 1149.1-Overlap-Zone concept with the 1149.1 portions shown on the left and the 1687-Only portion shown on the right; and with the Gateway elements being the only portion overlapping the two zones.

    - All instruments in the 1149.1 portion of the architecture, including the Gateway elements, are enabled by IR-Scans that install Instructions in the 1149.1-IR that become active on the falling-edge of **TCK** in the *Update-IR* state of the 1149.1-SM – and these instruments and their instructions are described by BSDL.

    - The 1687 portion of the architecture begins at the Gateway element or instrument – where the Gateway element or instrument enables other instruments to be accessed (by creating a select signal) and enables the passing of the TDI-TDO scan-path from the 1149.1 TAP to the target instrument in the 1687-Zone.

    - The Gateway is an element or instrument that is fully compatible with the 1149.1 sequence of operation, is connected to the 1149.1 TAP TDI-TDO serial scan-path, and supports hierarchical connection schemes to other instruments.

## #2: The Gateway

**This portion of the HW Proposal puts forth the concept that the 1687 hardware architecture (and language description) begins at a different point than the beginning point of the 1149.1 hardware architecture; but it does have a defined overlap and demarcation boundary with the 1149.1 hardware architecture; that this demarcation point can be defined and described; and that it can be verified, evaluated and checked for compliance in both the 1149.1 environment and the 1687 environment:**

***RULES:***

- This portion of the HW Proposal introduces the **required** concept of a Gateway or Gateway-Instrument as the beginning of the 1687-Only-Zone.

- …and **acknowledges** that all 1687 instruments that are not in the 1149.1-Overlap-Zone are accessed, controlled and configured only by DR-Scans (*Shift-DR* and *Update-DR* assertions) through a Gateway element or Gateway instrument.

- Only instruments that are accessed by the 1149.1-IR and support hierarchical-access to other instruments are **defined** as Gateways:

    - hierarchy in the 1687 sense is defined as an instrument that receives, at the very least, its “selection” and “serial scan-path” from another instrument or element other than the 1149.1 TAP and 1149.1-IR (alternatively, it can be viewed as moving instructions from the legal defined 1149.1-IR to other registers or elements);

    - instruments that are selected by the 1149.1-IR but do not support hierarchy are defined to be 1149.1-Overlap-Zone instruments;

    - and instruments that are connected to instruments that are not controlled directly from the 1149.1-IR are 1687-Only-Zone instruments.

- Gateways are **required** to be operable by a compliant 1149.1 TAP and 1149.1 Controller (must be a B-Type or C-Type instrument; not an A-Type or D-Type instrument) and must be describable by BSDL.

- …it is **preferred** that Gateways support a **RESET** Instruction (other than just an 1149.1 System-Reset using TRST* or the 5 **TCK**s while TMS=1 that places the 1149.1-SM into the *Test-Logic-Reset* state) to allow closing down of hierarchical-connections.

- …and Gateways (being 1149.1-Overlap-Zone elements) **must** be able to be enabled/accessed from an 1149.1-IR instruction and the **preferred** instruction format is encoded (as opposed to 1-hot) to minimize 1149.1-IR length.

- NOTE: even though one goal of 1687 is not to **require** anything that would make the 1149.1 architecture non-compliant, it must be noted that the 1687-Zone does not actually require a compliant 1149.1 architecture to operate successfully since the beginning of the 1687 hardware architecture is the Gateway. For 1687, it is not required that the 1149.1 Overlap-Zone portion of the architecture support a boundary scan register, the bypass register, an ID-code register, and many of the other standardized board-test features; or that it not be in a compliance enable mode – all that is needed to operate the Gateway and subsequent hierarchically connected 1687 instruments are:

    - the **TCK** and a **Reset** signal;
    - the Gateway Instruction in the TAP Instruction register;
    - the Capture-Shift-Update state-machine sequence,
    - the TDI-TDO serial scan-path.
    - this implies that 1687-compliance could be based on the 1687 architecture and may not require a compliant 1149.1 to be considered compliant (although this point has not been acceptance-voted yet).

    These considerations enable a core to claim 1687 compliance without mandating 1149.1 compliance. It is assumed that the device in which the core is finally embedded is 1149.1 compliant.

***Examples & Clarifications:***

- Figure 2 shows an example single cell, known as a Select-Instrument-Bit (SIB) that meets the needs of a Gateway element in that it can be an element in a Test-Data-Register (TDR) that is connected to the 1149.1-IR and can be used to pass hierarchical connections on to 1687 instruments:

    - the **Select-Instrument** (**Sel**) signal can be generated from an 1149.1-IR Instruction encoding as can the **Reset** (**Rst**) signal, but Rst must be ORed with the TAP TRST* or TLR 1149.1-SM state.

    - the **Shift-DR** *(***ShD**) enable signal can be generated from the 1149.1-SM when it is in the *Shift-DR* state and is applied when the **Sel** input is asserted.

    - the ShSIB register is the shift portion of the shift-update cell – and acts as a non-inverting pass-through bit when the cell is selected.

    - the TDI serial scan-path signal receives data from the previous element when the **ShD** signal is asserted, which is synchronized by the

**TCK**; and that data ultimately sources from the device-level TAP TDI pin.

- the **TCK** is the TAP-level synchronizing clock signal.

- the TDO serial scan-path signal produces data shifted by the ShSIB register to the next element when the **ShD** signal is asserted, which is synchronized by the **TCK**, and that data ultimately appears on the device-level TAP TDO pin.

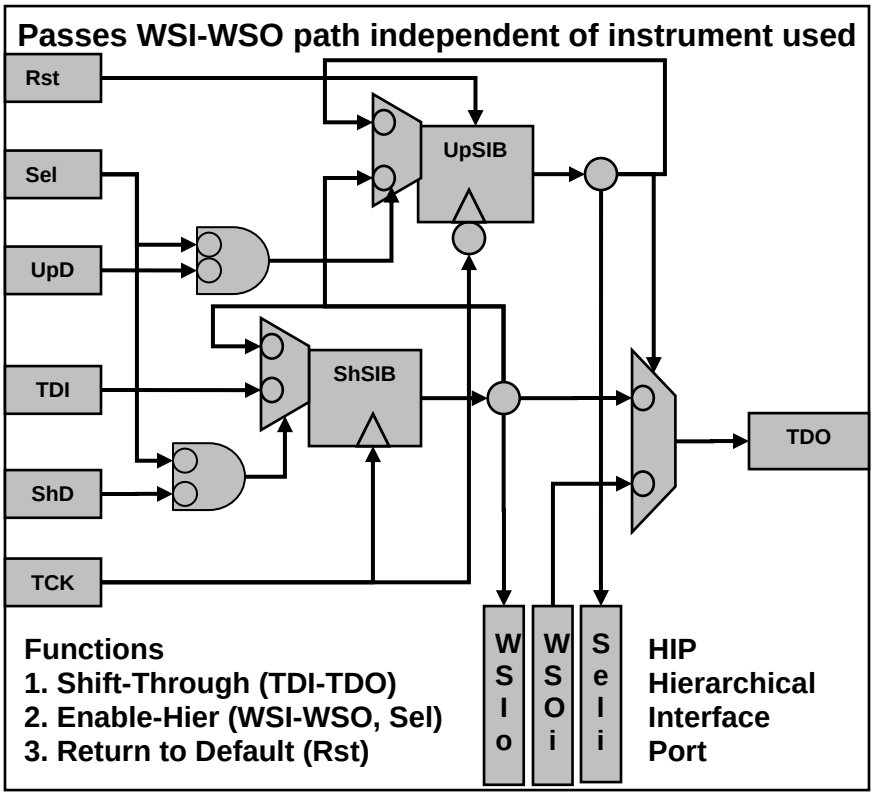


**Figure 2: The Select-Instrument-Bit Concept as a Gateway Bit**

- the **Update-DR** ***(*****UpD**) enable signal can be generated from the 1149.1-SM when it is in the *Update-DR* state and is applied when the **Sel** input is asserted.

- the UpSIB register is the update portion of the shift-update cell and acts as the enable to the Hierarchical Interface Port (HIP) when activated (when a logic 1 is registered by the falling edge of **TCK** while the **UpD** signal is asserted).

- activation of the HIP adds the WSI-output and the WSO-input to the TDI-TDO scan path after the ShSIB register and generates the static output **Select-Instrument pass-on *(*Seli**) signal.

- many of these SIBs can be concatenated TDI-to-TDO, with the control signals provided in parallel, to create a multi-bit TDR known as a Select-Instrument-Register (SIR) – where each bit in a one-hot fashion enables and connects TDI-TDO to an instrument, group of instruments, or further hierarchical connections.

- NOTE: a SIB may be comprised of more than one **ShSIB** and **UpSIB** signal pair to create additional configurable control signals associated with the HIP – for example, the **Select-WIR** signal needed to control a 1500 Core-Test Wrapper TAM or a locally-provided **Reset** signal.

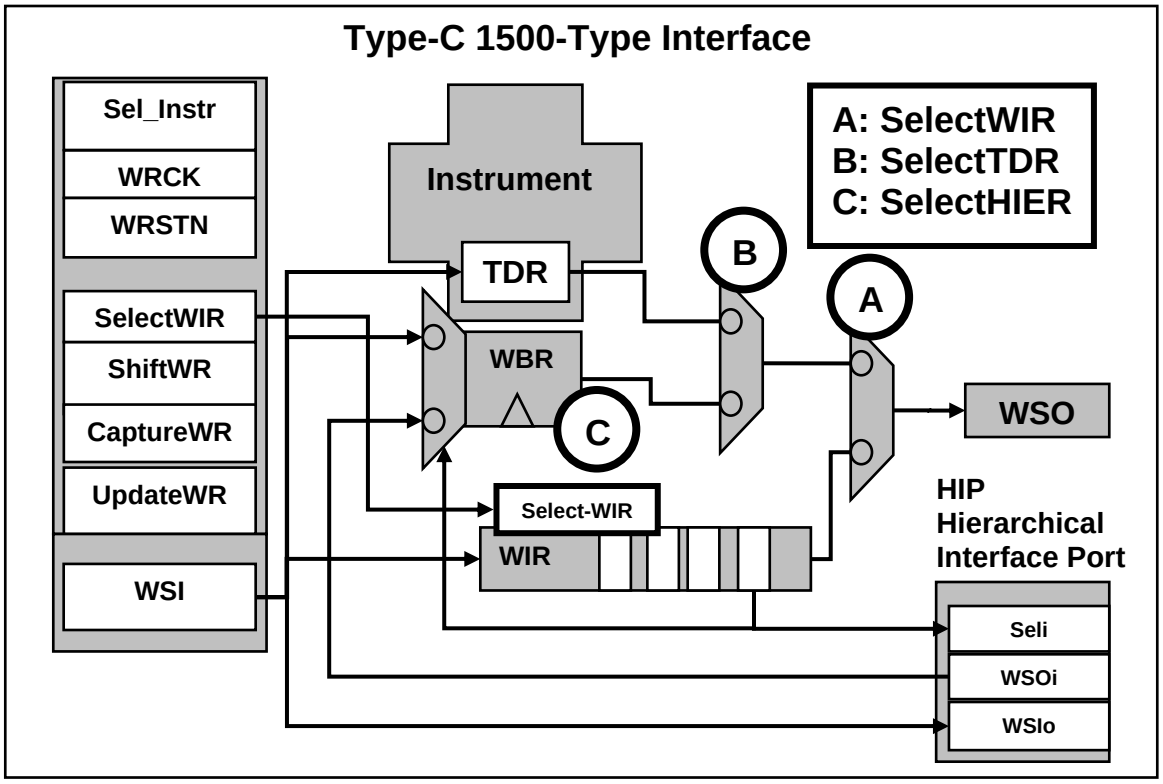


**Figure 3: An example 1500-TAM as a Gateway Bit**

- Figure 3 shows an example IEEE Standard 1500 type of Test-Access Mechanism (TAM) being used as a Gateway:

  - use as a Gateway is enabled when one or more of the instructions in the 1500 defined Wrapper Instruction Register (WIR) can enable one or more Hierarchical Interface Ports (HIPs).

- NOTE: the core or instrument that is coupled to the Gateway TAM is considered a 1687 instrument if it is selected, enabled, or configured completely from WIR instructions (one level away from the 1149.1-IR)

- NOTE: only the WIR, and potentially the Wrapper Bypass Register (if the WBY is the default configuration when the 1500-TAM is enabled by an 1149.1-IR Instruction), are considered 1149.1-Overlap-Zone instruments.

- the **WRCK**, **WRSTN**, **Shift-WR**, **Capture-WR**, **Update-WR** signals are provided by the TAP and 1149.1-SM as the **TCK**, **Reset**, **Shift**, **Capture**, and **Update** signals, respectively.

- the **Select-Instrument** (**Select-Instr**) and **Select-WIR** signals are provided by the 1149.1-IR by at least two different instructions (with **Select-Instr** being asserted by both and **Select-WIR** being asserted by one of them if the instructions are encoded).

- the TAM's WSI-WSO serial scan-path is connected between the TAP-level TDI-TDO when the TAM's **Sel-Instr** signal is asserted.

- the TAM's WIR is connected to the WSI-WSO serial scan-path when the **Select-WIR** input signal is asserted. This connection overrides any other scan-path connection, even the HIP.

- the selection of a Hierarchical Instruction in the TAM's WIR creates a **Seli** output signal and passes on the scan-path connection by connecting the WSIo-Output and WSOi-Input to the TAM's WSI-WSO serial scan-path.

- NOTE: other signals may optionally be passed on through the HIP – signals such as the **Reset** or the **Select-WIR** to the instrument connected to the HIP.

- NOTE: a WIR may support more than one instruction-bit to create additional configurable control signals associated with the HIP – for example, the **Select-WIR** needed to control another 1500 Core-Test Wrapper TAM or a locally-provided **Reset**.

- NOTE: multiple instructions may select or enable the same HIP – in some cases, the instructions may be complementary (such as separate instructions that create the **Reset** and **Select-WIR** signals for the same hierarchically-connected instrument), or the instructions may be mutually-exclusive (such as two different instructions that may connect different instruments to the WSOi and WSIo ports).

- NOTE: the serial path should pass through the WBR to manage timing – if the serial path is just passed on from the TAM without passing through the 1500-TAM, then the selection of an instrument through a hierarchical path with many elements can result in a long non-registered connection wire.

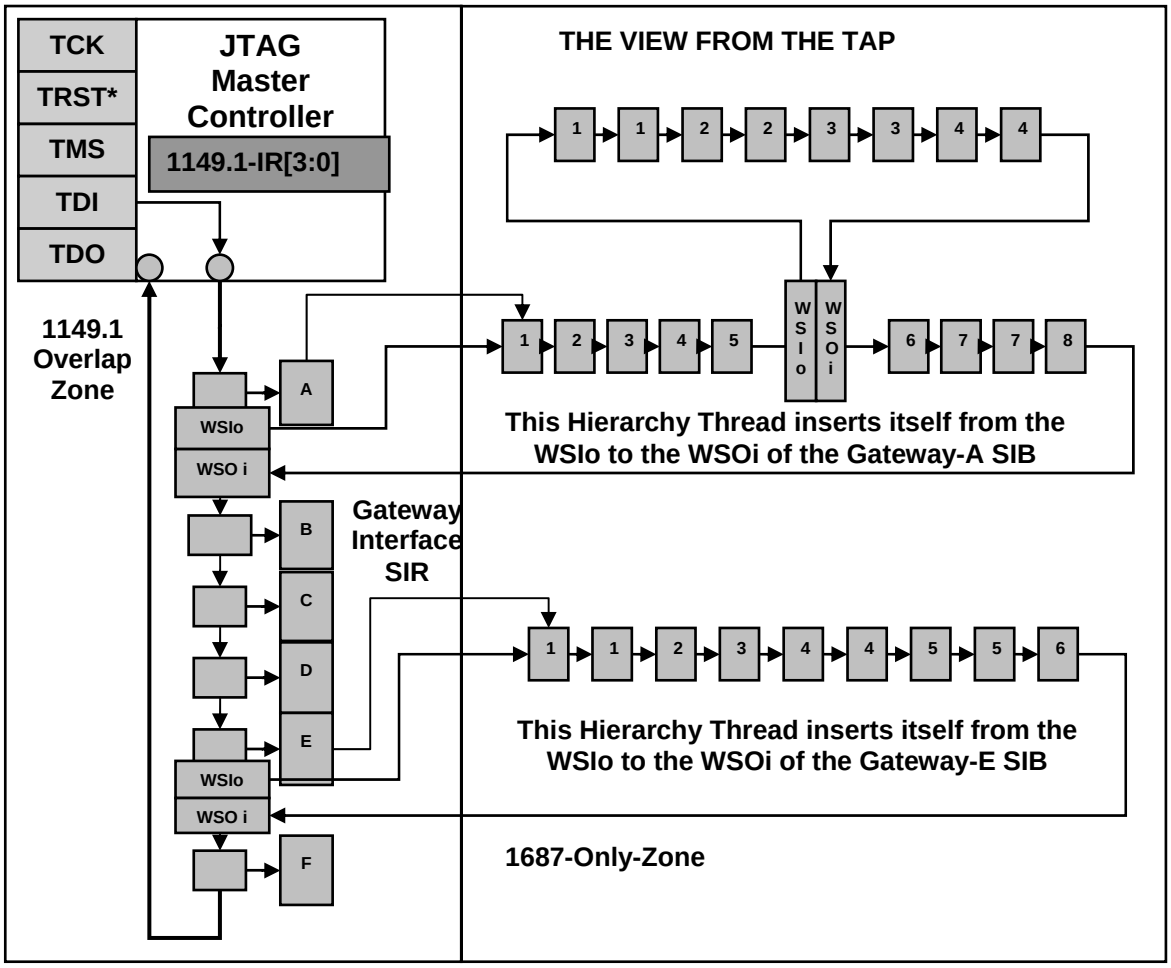


**Figure 4: An example Select-Instrument-Register used as a Gateway**

- Figure 4 shows an example of a grouping of Select-Instrument-Bits (SIBs) being used as a Select-Instrument-Register (SIR):
  - a collection of individual SIBs connected in-line TDI-to-TDO makes a Select-Instrument-Register (SIR).
  - the SIR is selected by an 1149.1-IR instruction (known as a Gateway-Enable instruction – GWEN) and when selected, the SIR's serial shift-path is connected to the TAP's TDI-TDO serial shift-path.

- the Gateway is described by BSDL – the shift-path size is represented as the default or reset value (there is currently no way to describe a dynamically changing shift-path in BSDL) and the definitions of the Update Bits are not necessary for 1149.1 since these will be described using the 1687 description language and since the 1687 instruments should not have anything to do with board-test as defined by the 1149.1 Standard.

- when an individual SIB goes through an *Update-DR* event to a logic 1 (bits shown with WSI-WSO ports), then the SIB enables its HIP.

- the HIP may connect to a single instrument; may connect to a group of instruments; or may connect to an instrument that itself has hierarchical connections to other instruments.

- Figure 5 shows an example of the beginning of the 1687 hardware architecture. Note that the 1687 hardware architecture starts at the Gateway register, not at the 1149.1 TAP. Even though it is 1687's goal to not impact the compliance of the 1149.1-Zone, the selection, configuration, and operation of the Gateway is possible from any source that can provide the signals and sequences shown.

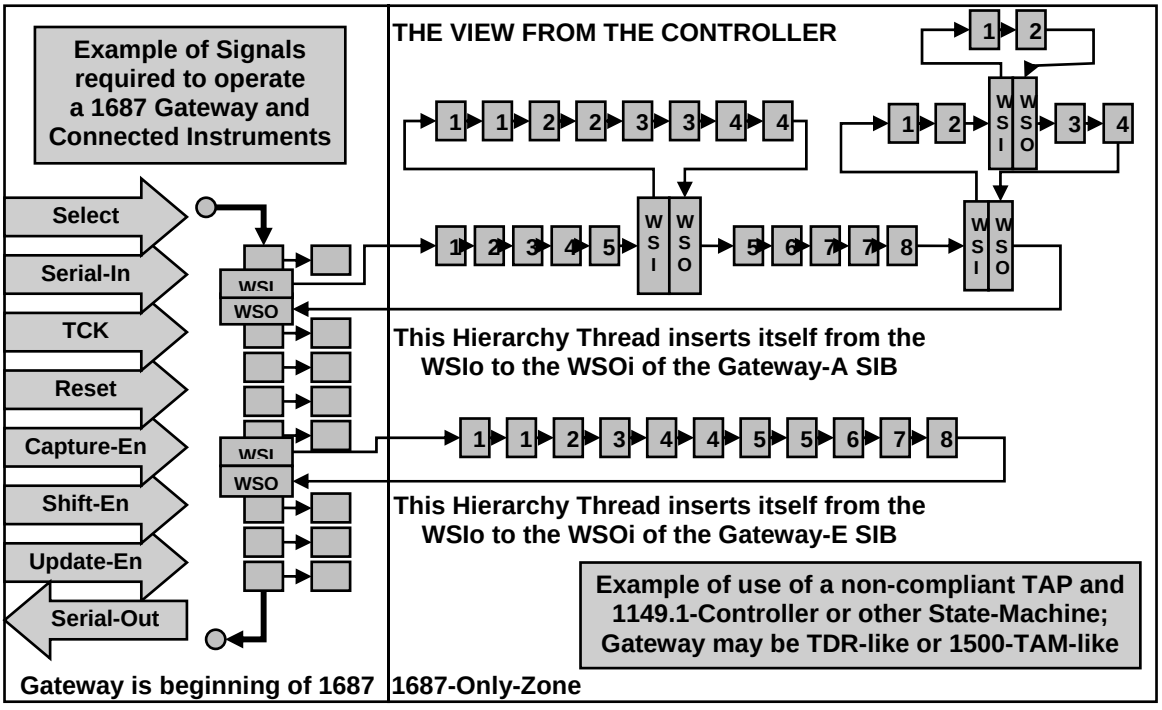


**Figure 5: An example of the 1687 starting at the Gateway**

- Figure 6 shows an example of two devices containing 1687 hardware architectures that are daisy-chained on a board – the 1687 architectures

look like one seamless architecture since the two Gateways (GWEN-1 and GWEN-3) effectively become one Super-Gateway that effectively make 2 devices into a single 1687 architecture.

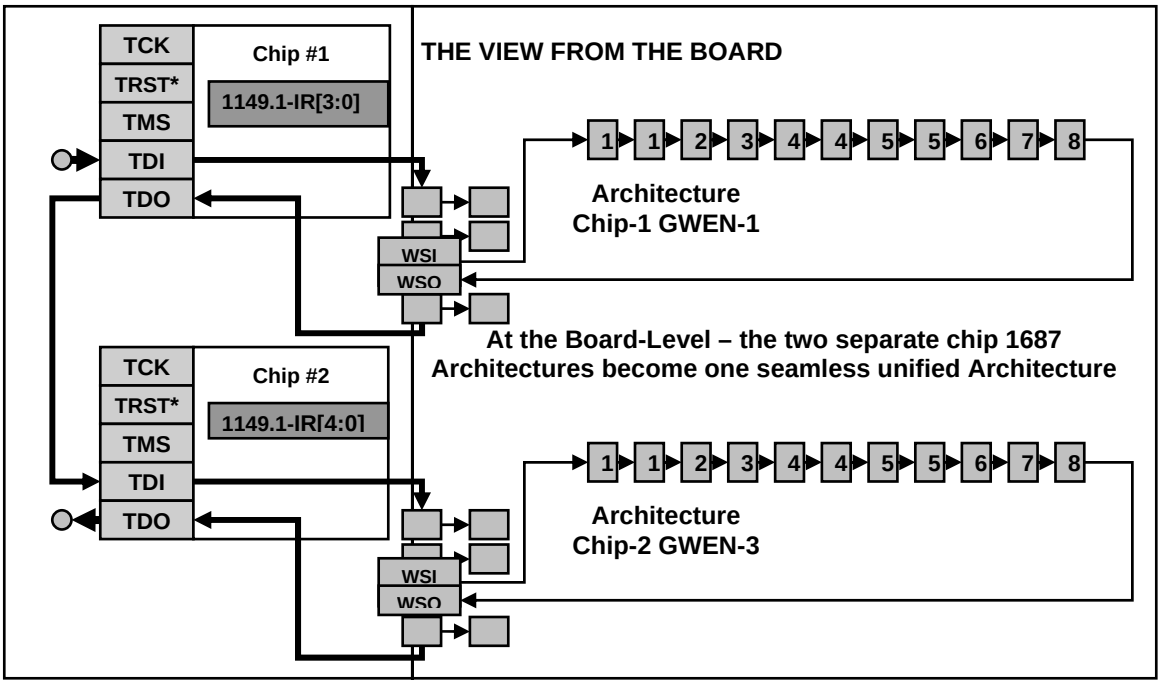


**Figure 6: Example of Two 1687 Devices Concatenated on a Board**

- NOTE: a similar device architecture as the ones shown with Gateways should result from using an 1149.1 TDR or a 1500-TAM as the Gateway Register except that the specific WSOi and WSIo connections may source from different locations in the Gateway serial shift-path (before or after the enable bit, or before or after the Gateway Registers).

## #3: The Archetypal Instruments – A, B, C, D

**In order to manage and describe the operation and connection of the instruments that will be defined to be 1687 instruments, there must be some limits placed on the interfaces. This portion of the HW Proposal puts forth the concept that there are 4 defined 1687 taxonomic instrument types, which should cover most common legacy instruments (including IEEE Standard 1500 based instruments) and drive the configuration of newly developed ones; that the instruments can be defined by their collection of signals and the operations they support; that the interfaces can be defined and described; and that they can be checked for compliance:**

***RULES:***

- This portion of the HW Proposal introduces the required concept of **limiting** the instruments to 4 Archetypal Instruments labeled A, B, C, and D (where instrument interface descriptions get more complex going from A to D).

- The Type-A instrument is **defined** as a "self-contained instrument" that is enabled by static-signals; and reports status by latched-output signals; has no serial-path; and supports no hierarchy (and hence, cannot be used as a Gateway).

  An example of a Type-A instrument is a simple Memory BIST controller.

- The Type-B instrument is **defined** as an "1149.1-compatible instrument" that operates identically to an 1149.1 defined TDR; has a serial-scan path; and may support hierarchy (and may be used as a Gateway with the caveat that the length of the TDI-TDO scan-path must be described in the BSDL of the 1149.1-Overlap-Zone as the default or reset length).

  An example of a Type-B instrument is any instrument that is directly managed by 1149.1 state-machine signals and associated Select-Capture-Shift-Update protocol.

- The Type-C instrument is **defined** as a "self-instructed instrument" that operates identically to an 1149.1 Compatible 1500-TAM that has multiple internal registers and requires a **Select-IR** signal; has a serial-scan-path that may have multiple paths or contributors, where one serial-scan-path is a local instruction register; and may support hierarchy (and hence, may also be used as a Gateway).

  An example of a Type-C instrument is a 1500-wrapped core with core boundary-scan cells that do not require the Transfer signal.

- The Type-D instrument is **defined** as a Type-B or Type-C instrument whose control interface supports at least one of the following: a signal or sequence not produced by a compliant 1149.1 TAP or 1149.1-Controller; a clock other than **TCK**; a data port other than the TDI-TDO serial scan-path (and hence, cannot be used as a Gateway since it is not easily described in BSDL – but it may still be used as a hierarchical instrument).

  An example of a Type-D instrument is a 1500-wrapped core with core boundary-scan cells that do require the Transfer signal.

| Port Name | Port Definition | Instrument | IWidth | Hierarchy | HWidth | Signal Type |
|---|---|---|---|---|---|---|
| **Select** | Enable the 1687 interface/instrument | Input | 1 | Output | n | Control |
| **InstrBits** | Static instruction input to the instrument | Input | n | Output | n | Control |
| **DataBits** | Static data input to the instrument | Input | n | Output | n | Data/Control |
| **TestReset** | Reset the 1687 interface/instrument | Input | 1 | Output | n | Control |
| **Status** | Indicate status or results from instrument | Output | n | | | Status |
| **TestClock** | Clock signal for the 1687 interface | Input | 1 | | | Clock |
| **ScanIn** | Serial scan data input to instrument | Input | 1 | Pass-Thru | 1 | Data |
| **ScanOut** | Serial scan data output from instrument | Output | 1 | Pass-Thru | 1 | Data |
| **CaptureEn** | Capture functional data from D-inputs | Input | 1 | | | Control |
| **ShiftEn** | Shift serial scan data | Input | 1 | | | Control |
| **UpdateEn** | Update shifted data to functional Q-outputs | Input | 1 | | | Control |
| **TAPStates** | Indicate state of IEEE 1149.1 TAP controller | Input | 1 each | | | Control |
| **SelectIR** | Select IEEE 1500 wrapper instruction register | Input | 1 | Output | n | Control |
| **ParallelIn** | Parallel data input clocked by TestClock | Input | m | | | Data |
| **ParallelOut** | Parallel data output driven w.r.t. TestClock | Output | n | | | Data |
| **AuxEn** | Enable alternate high-bandwidth data path | Input | 1 | | | Control |
| **AuxIn** | High-bandwidth data input | Input | m | | | Data |
| **AuxOut** | High-bandwidth data output | Output | n | | | Data |
| **AuxCtrlEn** | Transfer functional ports to AuxIn/Out control | Input | 1 | | | Control |

**Table 1: Super-Set of Instrument Signal Definitions**

- Instruments that are not compliant or compatible with the 4 Archetypes are **allowed** to be converted to a compliant instrument with interface conversions, logic, or state-machines.

- Instruments that are not compliant or compatible with the 4 Archetypes are **allowed** to exist and not be considered part of the 1687 architecture (not all instruments within a device are required to be part of either the 1149.1-Overlap-Zone or the 1687-Only-Zone).

- The allowed Instrument interface signals are named and **defined** in Table 1 (column 1, Port Name, and column 2, Port Definition) as the Super-Set of all possible allowed signals – the signal directions and whether the signal is allowed as a single or multiple group are given in columns 3, Instrument, and column 4, IWidth; the passing on of certain signals to support hierarchy, and their directions, and the sizing, which is related to the number of child instruments to be accessed, is given in columns 5, Hierarchy, and 6 HWidth; and the classification of the signal as Control, Data, or Clock is given in column 7, Signal Type.

| Signal | Type-A | Type-B | B-Hier | Type-C | C-Hier | Type-D | D-Hier |
|---|---|---|---|---|---|---|---|
| Use | Flat-Only | | Gateway | | Gateway | | Hier-Only |
| Select | Mandatory | Mandatory | Req. Pass | Mandatory | Req. Pass | Mandatory | Req. Pass |
| TestClock | Optional | Mandatory | NA | Mandatory | NA | Optional+ | Req.* |
| TestReset | Preferred | Optional | Optional | Mandatory | Optional | Optional | Optional |
| SelectIR | NA | NA | Optional | Mandatory | Optional | Optional | Optional |
| CaptureEn | NA | One of: | NA | Mandatory | NA | Optional+ | Req.* |
| ShiftEn | NA | One of: | NA | Mandatory | NA | Optional+ | Req.* |
| UpdateEn | NA | One of: | NA | Mandatory | NA | Optional+ | Req.* |
| InstBits | Preferred | Optional | NA | Optional | NA | Optional | Optional |
| Data Bits | Preferred | Optional | NA | Optional | NA | Optional | Optional |
| Status | Preferred | Optional | NA | Optional | NA | Optional | Optional |
| TAPStates | NA | Optional | NA | Optional | NA | Optional+ | Optional+ |
| ScanIn | NA | Mandatory | Req. Pass | Mandatory | Req. Pass | Optional+ | Req. Pass |
| ScanOut | NA | Optional | Req. Pass | Mandatory | Req. Pass | Optional+ | Req. Pass |
| ParallelIn | NA | NA | NA | Optional | Optional | Optional | Optional |
| ParallelOut | NA | NA | NA | Optional | Optional | Optional | Optional |
| AuxIn | NA | NA | NA | NA | NA | Optional | Optional |
| AuxOut | NA | NA | NA | NA | NA | Optional | Optional |
| AuxEn | NA | NA | NA | NA | NA | Optional | Optional |
| AuxCtrlEn | NA | NA | NA | NA | NA | Optional | Optional |

**Table 2: ABCD Instrument Signal Requirements**

- The mapping of the allowed Interface Signals to the Instrument Taxonomy (ABCD) are **defined** in Table 2 with the signal name given in column 1;

    - the Type-A instrument interface allowances are **defined** in column 2 with Mandatory, Optional, Preferred, and NA (not-applicable) directives – the column name is Flat-Only to indicate that the instrument does not support hierarchy and is not allowed to be a Gateway.

    - the Type-B instrument interface allowances are **defined** in column 3 with Mandatory, Optional, NA, and "One of" (indicating that one of the 3 signals **CaptureEn**, **ShiftEn**, or **UpdateEn** is Mandatory).

    - the Type-B instrument interface additional allowances to support Hierarchical use are **defined** in column 4 with Required-Pass-Through, Optional, or NA – the column name is Gateway to show that the Type-B is allowed to be Hierarchical and can be used as a Gateway.

    - the Type-C instrument interface allowances are **defined** in column 5 with Mandatory, Optional, and NA.

    - the Type-C instrument interface additional allowances to support Hierarchical use are **defined** in column 6 with Required-Pass-Through, Optional, or NA – the column name is Gateway to show that the Type-C is allowed to be Hierarchical and can be used as a Gateway.

    - the Type-D instrument interface allowances are **defined** in column 7 with Mandatory, Optional, and Optional+ (where the "+" indicates that the signal or sequence may be replaced by an **Aux** Signal).

    - the Type-D instrument interface additional allowances to support Hierarchical use are **defined** in column 8 with Optional, Optional+ (where the "+" indicates that the signal or sequence may be replaced by an **Aux** Signal), Required-Pass-Through, and Required* (where the "*" indicates that the signal or sequence is required to support Hierarchy) – the column name is Hier-Only to indicate that the instrument may support Hierarchy, but may not be used as a Gateway (since the inclusion of a non-1149.1 compatible signal or sequence makes it ineligible for operation or BSDL description in the 1149.1-Overlap-Zone).

- Instruments are **allowed** to configure and control high-bandwidth ports which can be routed internal to the device; or may exist at the boundary of the device (I/O); and are **defined** as: parallel digital signals that synchronize to the **TestClock**; parallel digital signals that synchronize to an **AuxClock**; high-speed digital serial-signals that synchronize to an **AuxClock**; or high-speed

non-digital signals that must pass through a signal-conversion (such as a Parallel-to-Serial SerDes convertor).

- Note: Instruments are **not required** to be exclusively controlled by 1687. Other access, configuration, and control mechanisms are allowed (for example, a parallel bus-based access and update to the parallel pins of a TDR being used as a Gateway would allow 1687 instruments to be accessed from an alternate control source).

***Examples & Clarifications:***

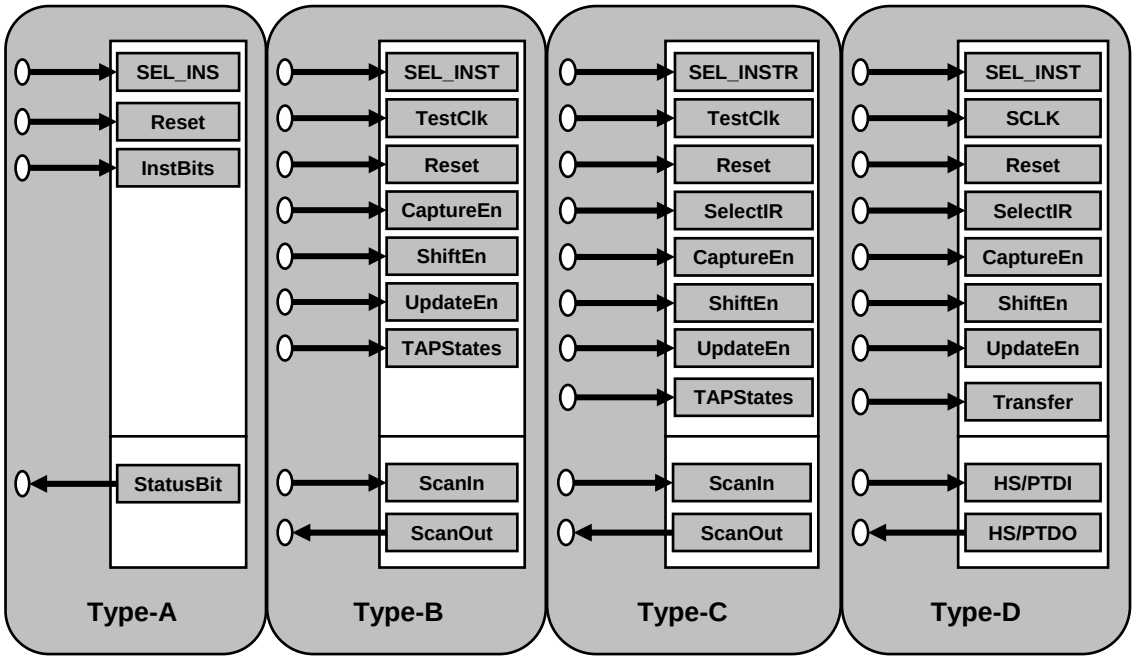


**Figure 7: Example ABCD-Taxonomy Instrument Interfaces**

- Figure 7 shows an example of each of the A, B, C, and D non-hierarchical instrument interfaces and their connectivity directions.

  - The Type-A interface requires the **Select-Instrument** signal; and has other static inputs such as **Invoke** and **Mode** and other signals that can be considered to be Instruction-Bits from an Instruction-Register; a static **Reset** input; and latched or "sticky" output signals such as **Done**, **Pass**, **Fail**, **Error**, and other signals that can be considered to be Status-Bits. The Type-A instrument interface is expected to be used on "self-contained" instruments such as Logic-BISTs, Memory-BISTs, and other standalone type logic units. Since the communication is with static signals that are applied on the 1149.1 *Update-DR* and sampled by the 1149.1 *Capture-DR*, as opposed to the clocked-operation of a local scan register,

the clock that actually operates the instrument is not required to be part of the instrument interface.

- The Type-B interface also requires the **Select-Instrument** signal; requires the **Test-Clock** to operate the serial shift-path; should support the **Reset** signal to keep the instrument quiescent when not used; the **Shift-Enable** signal and either the **Capture-Enable** and/or the **Update-Enable** needs to be supported; sometimes other 1149.1-SM-generated signals are required such as those generated in the *Run-Test-Idle* or *Pause-DR states*; and the serial shift-path TDI and TDO must be supported. The Type-B instrument is one that operates like an 1149.1 defined Test Data Register (TDR) – a register that has a shift-path that is active when the 1149.1-SM is in *Shift-DR*; and the shift-path may sample targeted signals when the 1149.1-SM is in *Capture-DR*; and optionally, control bits and functions that are applied when the 1149.1-SM is in *Update-DR*.

- The Type-C interface also requires the **Select-Instrument** signal; requires the **Test-Clock** to operate the selected serial shift-path; should support the **Reset** signal to keep the instrument quiescent when not used; the **Shift-Enable** and **Update-Enable** signals and optionally the **Capture-Enable** needs to be supported; sometimes other 1149.1-SM-generated signals are required such as those generated in the *Run-Test-Idle* or *Pause-DR* states; and the serial shift-path TDI and TDO must be supported. The Type-C instrument is one that operates like an 1500-defined Test Access Mechanism (TAM) – a set of registers, one of which is active as a shift-path that is active when the 1149.1-SM is in *Shift-DR* and based on the instructions in a register defined as an Instruction-Register; and the shift-path may sample targeted signals when the 1149.1-SM is in *Capture-DR*; and control bits and instructions that are applied when the 1149.1-SM is in *Update-DR*. The main interface difference from a Type-B is the **Select-WIR** signal that is used to unconditionally select the local instrument Instruction Register.

- The Type-D interface can be identical to the Type-B or Type-C, but must have at least one non-1149.1 compatibility issue: either the instrument requires a clock in addition to or other than **Test-Clock** (the 1149.1 **TCK**) to operate some portion of the interface; the instrument requires a data path other than the defined TDI-TDO serial shift-path that synchronizes to the **Test-Clock** (such as a Parallel-Port that synchronizes to **Test-Clock**; or a Parallel-Port that synchronizes to an **Alternate-Clock**; or a High-Speed Serial-Port that synchronizes to a High-Speed **Alternate-Clock**); the instrument requires a signal not provided from a compliant 1149.1 Controller or 1149.1-SM (such as **Stall**, **Bus-Reques**t, **Data-Valid, Counter-Done**, etc.); or the instrument requires a sequence not provided by a compliant 1149.1 Controller or 1149.1-SM (such as the 1500-defined Transfer function which falls between *Capture-DR* and *Shift-DR*). The

Type-D instrument is expected to be an instrument such as a complex 1500 Core-Test Wrapper that supports Transfer, or an item such as a Bus-Controller, Bus-Converter, or Clock-Controller that can be configured and controlled through the 1687 architecture.

- Non-Hierarchical instruments are either connected to the 1149.1 TAP and 1149.1 Controller in the 1149.1-Overlap-Zone (and are therefore 1149.1 Instruments) or they can be connected in the 1687-Only-Zone as non-hierarchical instruments that are the "end-stubs" of an instrument connectivity scheme (meaning that no further hierarchical connections are possible from these instruments).

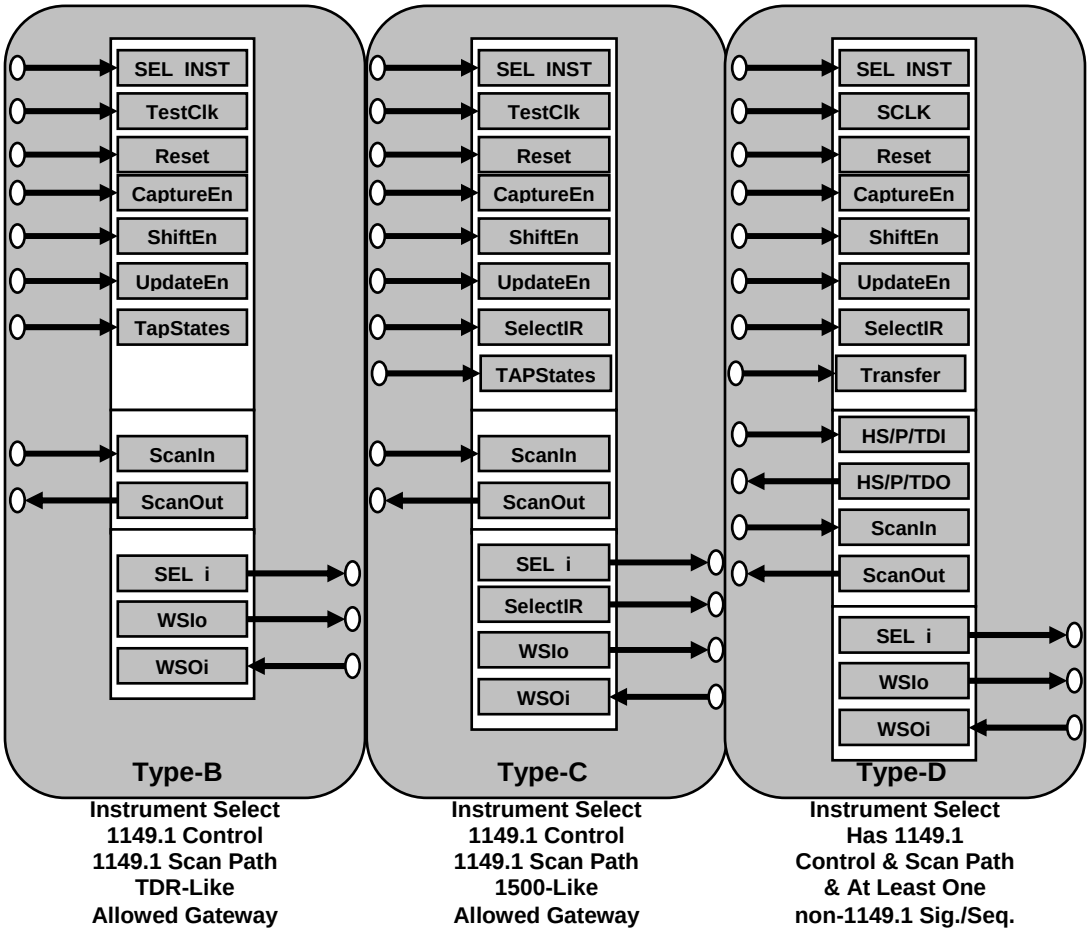


**Figure 8: Example Hierarchical Instrument Interfaces**

- Figure 8 shows examples of the allowed hierarchical instrument interface – note that only the Type-B and Type-C are allowed to be Gateway elements since the Type-D by definition includes elements that may not be operable by a compliant 1149.1 TAP-Controller; and/or may not be describable by BSDL.

- Examples of Instrument-Conversions are given in Appendix B. Conversions are the changing of one instrument interface to another and can be the

conversion of a defined 1687 interface of one type to another – such as turning a Type-A into a Type-C that supports hierarchy by adding a TDR or SIR or even a single SIB; or turning a Type-D 1500 TAM that supports a non-1149.1-compatible-sequence into a Type-B by placing a TAP-Controller-like State-Machine between a 2-Bit SIB and the Type-D interface. A conversion can also be the conversion of a completely non-1687 instrument to a 1687 by placing logic between the TDR, SIR, TAM, or SIB and the natural instrument interface.

- Instruments support hierarchy by supporting a Hierarchical-Interface-Port (HIP) – a minimal HIP is comprised of **Sel_i**, **WSIo**, and **WSOi**. Other HIP select-type signals are allowed such as **Reset**, **Select-WIR**, **Clock-Enable**, and **Pause**.

## #4: The Instrument Connectivity Schemes

**This portion of the HW Proposal puts forth the concept that there are 2 basic connectivity-types (flat and hierarchy) for 1687 instruments; and that there are several schemes under each type that can be used to optimize various engineering and economic tradeoffs; and that defining and limiting the connection-schemes allows compliance checking:**

***RULES:***

- The connection-scheme of the instruments, including Gateways, in the 1149.1-Overlap-Zone **shall** meet the rules, requirements, tradeoffs, and optimizations allowed by the 1149.1 Standard and Board-Test use.

- The connection-scheme of the instruments, from the Gateway outward (the instrument side of the Gateway) to the first-layer of instruments in the 1687-Zone **shall** be limited to the 3 hierarchy-connection-schemes: replace; concatenate-before; or concatenate-after.

- The connection-scheme of the instruments fully within the 1687-Zone **shall be limited** to the 3 hierarchy-connection-schemes and the 4 flat-connection-schemes (Flat, Daisy-Chain, Star, and Concatenate) – the preferred schemes should meet the needs of engineering & economic-utility tradeoffs **defined** as:

**Engineering Tradeoffs:**

- **Area-Impact**: the amount of logic added to implement connectivity, hierarchy or concurrence.

- **Routing-Congestion**: the amount of wiring required to implement a connectivity scheme.

- **Power**: the amount of logic required to be active when a particular connectivity scheme is applied.

- **Risk**: the sensitivity of the connection scheme to faults, failures, errors or conflicts (such as, for example, a bad bit in a daisy-chain connection breaks all access to all instruments).

**Operating Efficiency Tradeoffs:**

- **IR-Width**: the number of bits in the 1149.1-IR that affects the board-test operation and efficiency of the 1149.1-Overlap-Zone.

- **Scan-Path Depth**: the bit-length of the 1149.1 TDI-TDO scan-path as instruments are accessed.

- **Scan-Path Depth-Stability**: the probability that the scan-path depth will change bit-length and bit-definitions while under an 1149.1 Instruction.

**Utility/Automation Tradeoffs:**

- **Concurrence**: the ability for multiple instruments to be operated simultaneously.
- **Flexibility**: the ability to select and configure multiple instruments once the architecture is hardened in silicon.
- **Protocol-Complexity**: the extra required number of steps or sequences needed to access and operate instruments.
- **Language-Complexity**: the extra information required to describe the instrument interface and connectivity scheme.

- A Hierarchical-Connection is **defined** as: the **Select** signal sourcing from an instruction not in the 1149.1-IR, coupled with passing-on the TDI-TDO scan-path connection – the passing of the TDI-TDO must include the concept of bypass registration to guard against long TDI-TDO wires with no timing control.

- Hierarchy is **allowed** to be supported from within an instrument or bolted-on to the outside of the instrument by using an element such as the Select-Instrument-Bit (SIB).

- The **preferred** hierarchy-support at the instrument should include localizing the instrument select signals (**Select**, **Select-IR**) to reduce the Protocol-Multiplication problem.

- NOTE: The **preferred** result is that any 2 1687-devices connected TDI-to-TDO should seem to act as a single unified 1687 architecture (see Figure 6).

- NOTE: More clarification of the various connection schemes against the listed tradeoffs can be found in Appendix-C.

***Examples & Clarifications:***

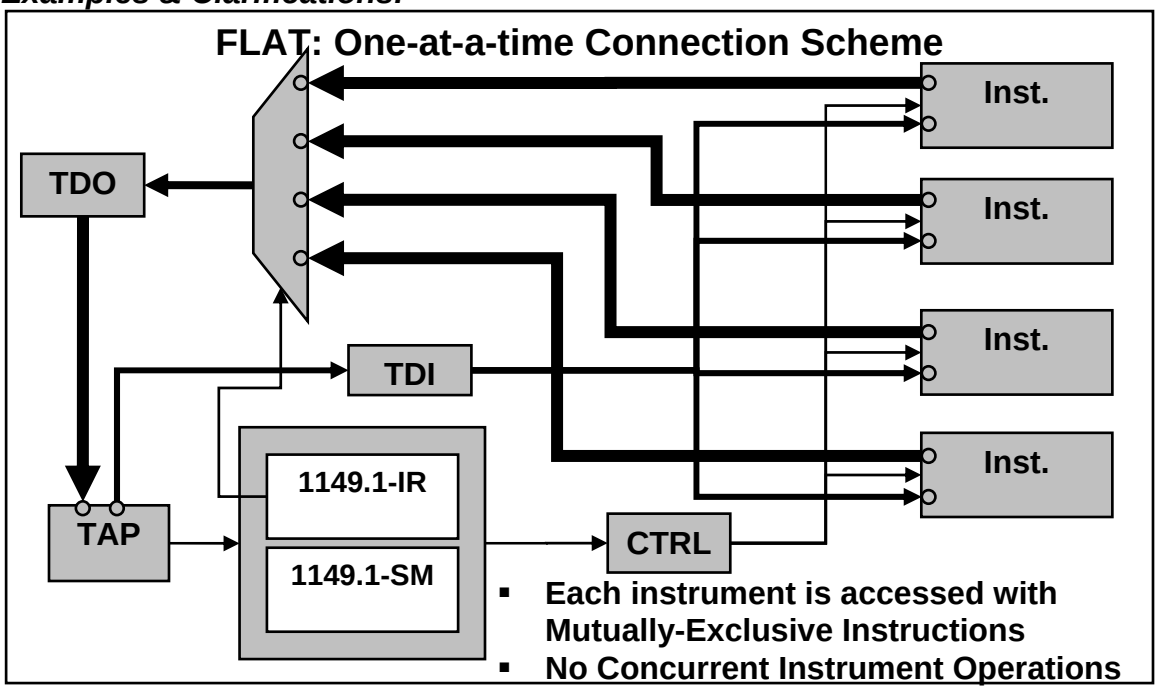


**Figure 9: Example Flat-Connection**

- Figure 9 shows an example of the FLAT connection scheme where an Instruction Register (shown here as the 1149.1-IR in the 1149.1-Overlap-Zone, but this concept can also be applied to any distributed Instruction-Register in the 1687-Only-Zone) contains mutually exclusive instructions that can only enable one instrument at a time and connect only that one instrument to the 1149.1 TDI-TDO scan-path.

  - A negative tradeoff is no CONCURRENCE and no FLEXIBILITY – in that no simultaneous instrument operation is possible.

  - A negative tradeoff is excessive IR-WIDTH – in that each instrument requires one instruction.

  - A negative tradeoff is more ROUTING-CONGESTION – in that it requires a separate routing from the TAP-Controller to each instrument.

  - A positive tradeoff is short SCAN-PATH-DEPTH that is STABLE – in that it does not dynamically change while an instruction is active.

  - A positive tradeoff is less RISK – in that a broken scan path may only disable one instrument.

  - A positive tradeoff is less POWER – in that only one instrument is active at any given time.

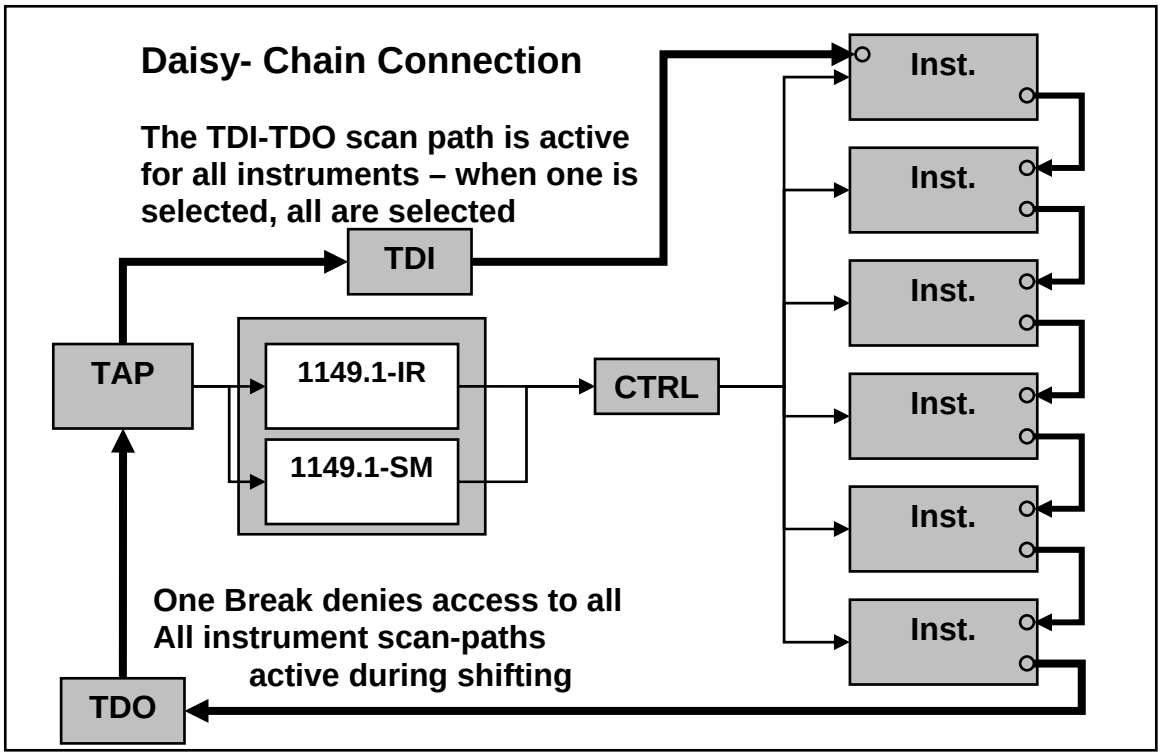


**Figure 10: Example Daisy-Chain-Connection**

- Figure 10 shows an example of the DAISY-CHAIN connection scheme where an Instruction Register (shown here as the 1149.1-IR in the 1149.1-Overlap-Zone, but this concept can also be applied to any distributed Instruction-Register in the 1687-Only-Zone) may contain many instructions that can activate any given instrument – or may contain just one instruction to activate all instruments. However, as soon as one instrument is enabled, all instruments are connected to the 1149.1 TDI-TDO scan-path.

    - A negative tradeoff is more RISK – in that one broken scan-path bit defeats access to all instruments.

    - A negative tradeoff is more POWER – in that all instruments scan-paths are active simultaneously.

    - A negative tradeoff is less FLEXIBILITY – in that all instruments scan-paths are active simultaneously which allows no other selection options.

    - A negative tradeoff is long SCAN-PATH-DEPTH – in that all instruments are in the serial scan-path making for the longest shift-path.

    - A positive tradeoff is a more SCAN-PATH-DEPTH STABILITY – in that it does not dynamically change while an instruction is active.

- A positive tradeoff is small IR-WIDTH – in that one instruction can enable all instruments.

- A positive tradeoff is less ROUTING-CONGESTION – in that it requires a routing from instrument-to-instrument, not separate routing from the TAP-Controller to each instrument.

- A positive tradeoff is more CONCURRENCE – in that all instruments control and shift-paths may be active and therefore each instrument can be active simultaneous to other instruments.

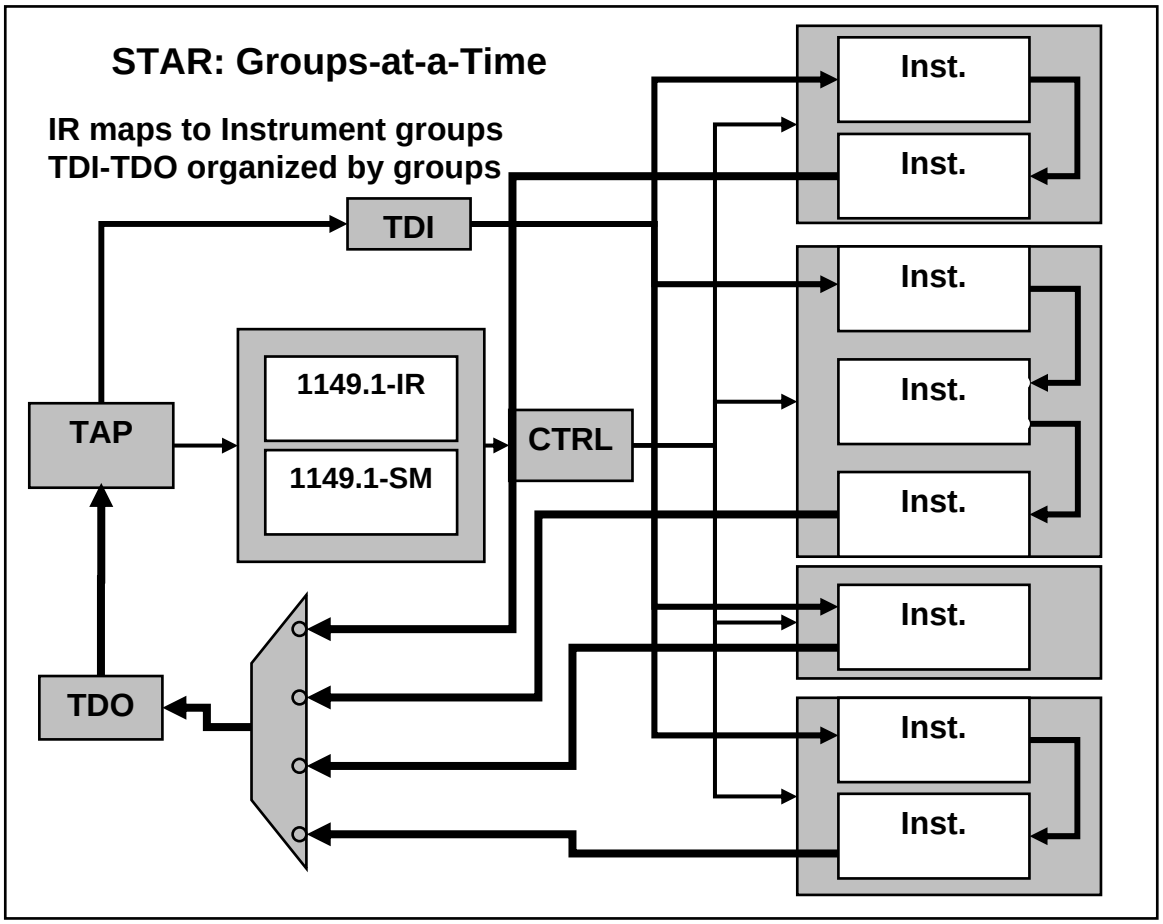


**Figure 11: Example Star-Connection**

- Figure 11 shows an example of the STAR connection scheme where an Instruction Register (shown here as the 1149.1-IR in the 1149.1-Overlap-Zone, but this concept can also be applied to any distributed Instruction-Register in the 1687-Only-Zone) may contain many instructions that are mutually exclusive, but can activate any given instrument or groups of instruments. This connection scheme is a mixture of FLAT and DAISY-CHAIN, but alleviates the negative tradeoffs identified with both.

- CONCURRENCE can be supported by grouping instruments that must be used simultaneously.

- RISK is alleviated by having groupings – a bad scan-path bit in an instrument only denies access to a group of instruments.

- POWER consumption is alleviated because all instruments are not active simultaneously.

- The perceived negative tradeoff for this scheme is that the architecture must be planned and implemented with the groupings decided and fixed during the architecture design phase – there is little other FLEXIBILITY once hardened.

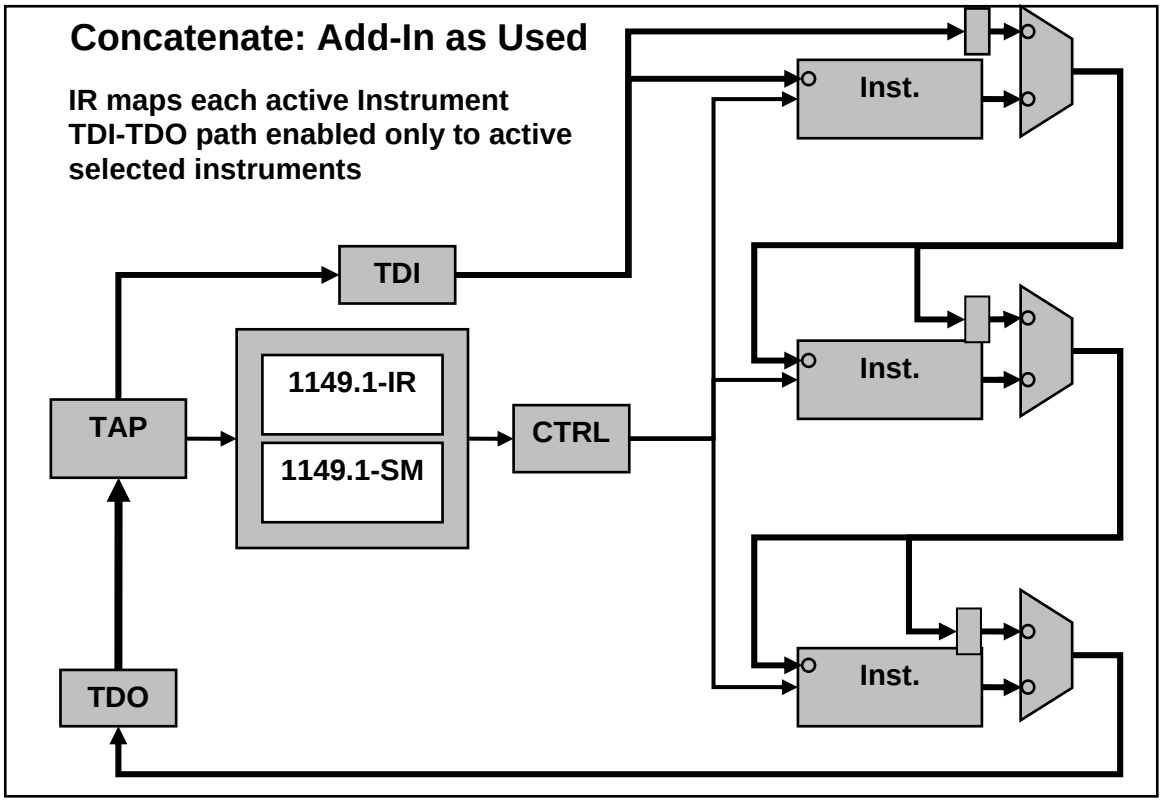


**Figure 12: An example Concatenate-Connection**

- Figure 12 shows an example CONCATENATE connection scheme where an Instruction Register (shown here as the 1149.1-IR in the 1149.1-Overlap-Zone, but this concept can also be applied to any distributed Instruction-Register in the 1687-Only-Zone) may contain many instructions that are not-mutually exclusive (for example, a one-hot instruction-bit scheme). Each time an instrument is selected, that instrument is added into the overall TDI-TDO scan-path – when not selected the instrument is locally bypassed (and the bypass should be through an instrument-local Bypass Register). This connection is similar to the Daisy-Chain in that all instruments are connected, but their active scan-paths are not required.

- A negative tradeoff with this scheme is more connectivity and timing RISK – in that it involves the concept of wire-connects from instrument-to-instrument. The RISK depends on the support of the Bypass-Register – if there are a large number of instruments connected mux-to-mux and they did not support a Bypass-Register at each instrument connection, then a long interconnecting non-timing-managed wire may result. If the Bypass-Register is supported, and there are a large number of instruments, then the default minimum scan-path may include as many bits as there are instruments.

- A negative tradeoff is excessive IR-WIDTH – in that this scheme is best supported with one-hot instructions and each instrument requires one instruction.

- A negative tradeoff is more AREA – in that there are required multiplexors between each instrument connection to the next instrument.

- A negative tradeoff is a more dynamic SCAN-PATH-DEPTH-STABILITY – in that the scan-path does change while an instruction is active it this connection scheme is used in the 1687-Only-Zone.

- A positive tradeoff is less ROUTING-CONGESTION – in that it requires a routing from instrument-to-instrument, not separate routing from the TAP-Controller to each instrument.

- A positive tradeoff is the shortest active SCAN-PATH-DEPTH – in that it is only as long as the active selected instrument scan-paths.

- A positive tradeoff is more CONCURRENCE – in that simultaneous instrument operation is easily managed.

- A positive tradeoff is more FLEXIBILITY – in that selection and configuration of multiple instruments for simultaneous operation is easily managed.

- A positive tradeoff is less failure RISK – in that a broken scan path may only disable one instrument.

- A positive tradeoff is less POWER – in that only selected instruments are active at any given time.

- NOTE: Hierarchical connections are very similar to the CONCATENATE connection, but the subtle difference is that the instrument that would normally be involved in the CONCATENATE connection instead has an

alternate local-IR-enabled connection that allows it to support its own grouping of CONCATENATE connections. This has the effect of hiding instruments or groups of instruments from the serial scan-path (with or without Bypass-Registers) until they are activated.

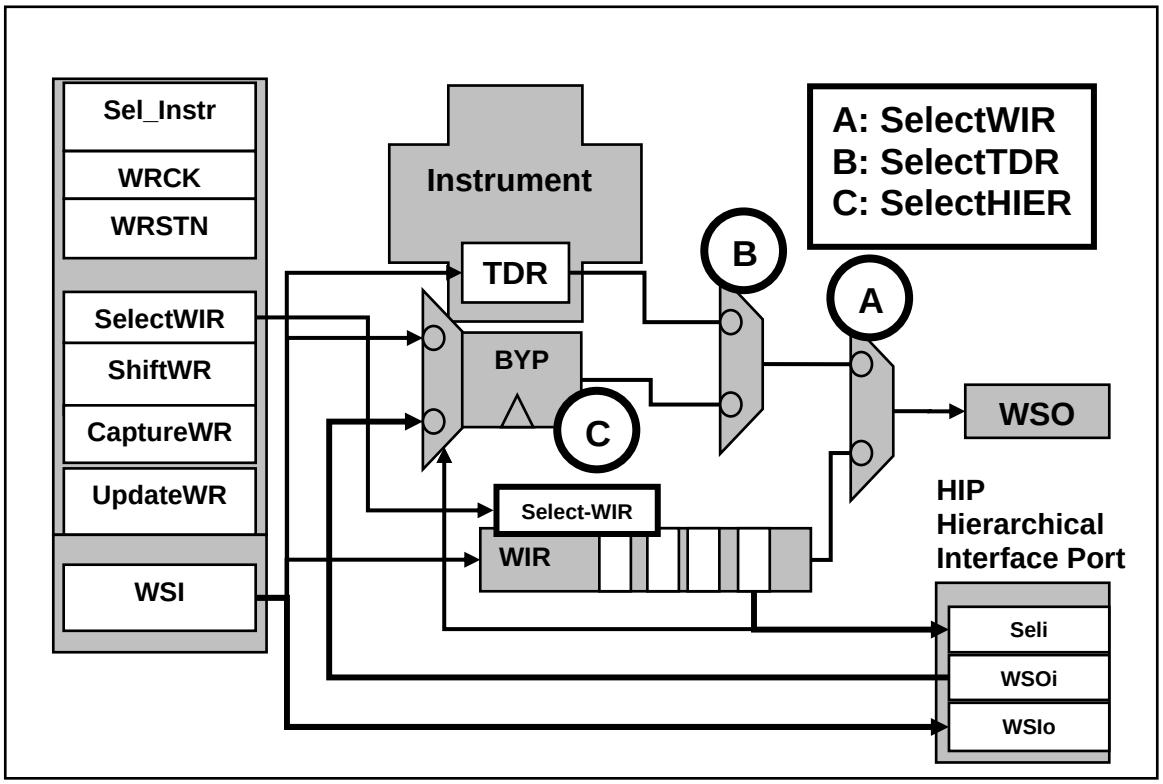


**Figure 13: An example 1500-TAM with Replace-Hierarchy**

- An example of a HIERARCHICAL connection scheme is shown in Figure 13 (as well as in Figures 2 and 3 involving the Gateway elements and instruments). The example shown in Figure 12 is a “REPLACE” connection scheme since the instrument itself can not use the WSI-WSO scan-path when the HIP is using it through the Bypass-Register.
    - The negative tradeoff for REPLACE is that using this type of connection is an either-or – either use of the instrument or use of the HIP, not both (which is good if default Mutual-Exclusivity is required).
    - Hierarchical connections have the positive tradeoff of minimal ROUTING-CONGESTION – in that Select-Instrument signals and the serial scan-paths source from the hierarchical instrument, not the centralized 1149.1 TAP-Controller.

- Hierarchical connections have the positive tradeoff of CONCURRENCE and FLEXIBILITY – in that multiple instruments may easily be activated to operate simultaneously.

- Hierarchical connection POWER and SCAN-PATH-DEPTH tradeoffs depend on the depth of the hierarchy supported – in that each level of hierarchy may require the Parent-Instruments to remain active.

- Hierarchical connections result in the off-loading of instructions from the 1149.1-IR to the Local Instrument-IR and therefore result in smaller IR-WIDTHs being possible.

- Hierarchical connections, however, represent negative tradeoffs for SCAN-PATH-DEPTH-STABILITY, PROTOCOL-COMPLEXITY, and LANGUAGE-COMPLEXITY – in that the required description, operation, and management of dynamically changing scan-paths from distributed instructions is a complex proposition.

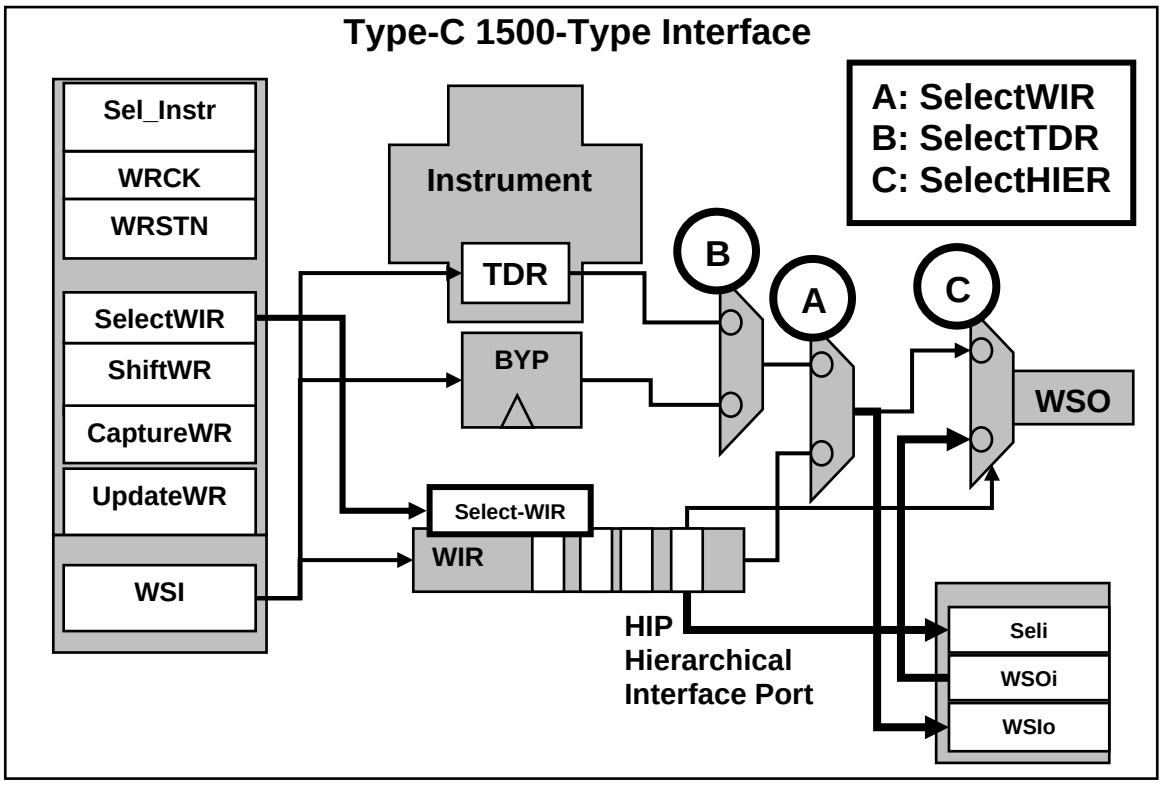


**Figure 14: An example 1500-TAM with Concatenate-After-Hierarchy**

- Figure 14 shows a CONCATENATE-AFTER hierarchical connection scheme where the HIP is applied after the instrument.

  - If the instrument is not actively being used, then the Bypass-Register is applied to the scan-path before the HIP;

- if the instrument is active, then the TDR is in the scan-path before the HIP;
- if the WIR is selected, then the WIR may be applied before the HIP, or the HIP may be removed from the scan-path while the WIR is active;
- if the HIP is not selected, then the selected instrument register passes directly from the WSI to the WSO.

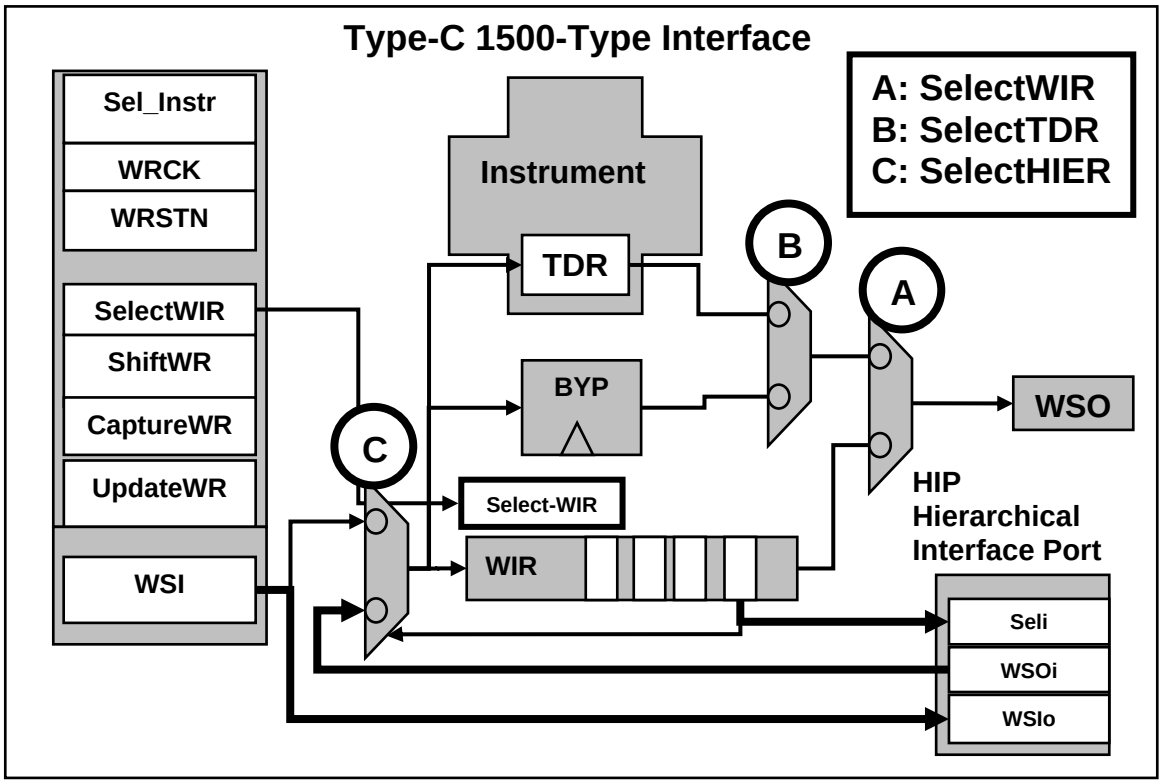


**Figure 15: An example 1500-TAM with Concatenate-Before-Hierarchy**

- Figure 15 shows a CONCATENATE-Before hierarchical connection scheme where the HIP is applied before the instrument.
  - If the instrument is not actively being used, then the Bypass-Register is applied to the scan-path after the HIP;
  - if the instrument is active, then the TDR is in the scan-path after the HIP;
  - if the WIR is selected, then the WIR may be applied after the HIP, or the HIP may be removed from the scan-path while the WIR is active; if the HIP is not selected, then the selected instrument register passes directly from the WSI to the WSO.

## The Loose Ends

**This portion of the HW Proposal lists the elements that are not specified by this proposal, but are items that will need to be considered in the future. It is expected that these items will be revisited, analyzed and eventually specified as a result of the language effort:**

- Asynchronous communication between instruments.
- Triggers, Breakpoints, and other embedded synchronization elements.
- The exact nature of the Bandwidth Interface control and configuration.

## The HW and the Language

**Here are some comments about this proposed Hardware and the Description and Protocol Languages:**

- One purpose of the hardware description is to narrow or limit the architecture that needs to be described – the purpose of this HW Proposal is exactly that, to explain the limit of what is allowed so the language effort can be applied to a known Strawman-Architecture.
- The Gateway concept allows the beginning of the 1687 architecture to be the Gateway element, not the 1149.1 TAP, and so all of the 1687 instruments are hidden behind the Gateway's Update-Bits.
- The Gateway concept also means that all actions taken against the 1687 instruments, no matter what the instrument-type or the connectivity-scheme, are coordinated only by 1149.1 DR-Scans (*Shift-DR* and *Update-DR* synchronization events).
- The only items requiring 1149.1 IR-Scans are the Gateways and they and the 1149.1 Instructions that select them are described as 1149.1-Overlap-Zone items with BSDL.
- This allows the 1687 description and protocol languages to be simplified to just handling items not in the 1149.1-Overlap-Zone.
- Unfortunately, this might lead to a description and protocol language problem or perturbation – there might be a need to deal with 1687 instruments that are in the 1149.1-Overlap-Zone which means that some of the Overlap-Zone may need to be described, and that the concept of 1149.1 IR-Scans may need to be handled by the languages.

# Glossary of Terms, Acronyms, and Abbreviations

There are many terms, acronyms and abbreviations used in this document that may be unfamiliar to the reader. A Glossary organized hierarchically by Standard is provided.

**IEEE 1149.1-2001**

- the board-test standard that defines an on-chip test-access-port (TAP) of 4, optionally 5, pins and associated signals and mandatory on-chip logic (Instruction Register, 16-stateTAP Controller, Bypass Register and Boundary-Scan Register) that is to be used to verify device connectivity in the board-environment. The standard is sometimes synonymously referred to as JTAG – see below.

**1149.1 Controller**: the logic attached to the TAP that includes the TAP State-Machine (1149.1-SM) and the TAP Instruction-Register (1149.1-IR) and its decode logic.

**1149.1-IR**: the acronym for the 1149.1 Instruction-Register which is a shift register that installs defined public and private instructions on the falling-edge of **TCK** when the TAP State-Machine is in the Update-IR state.

**1149.1-SM**: the acronym for the 16-state 1149.1 State-Machine which when coupled with instructions in the 1149.1-IR provides control signals for the connected BSR, Bypass Register, and other TDRs; and for the 1149.1-IR.

**Board-Test**: the verification that the devices connection to the board is intact by use of on-chip logic that can drive output pins to known logic values; can establish the direction of bidirectional pins; can establish the high-impedance state on three-state pins; and can sample logic values applied to input pins.

**Boundary-Scan**: the use of a serial scan-shift register called the Boundary-Scan Register (BSR) to provide controllability and observability of device I/O pins.

**BSDL**: the boundary-scan description-language – a VHDL-based language associated with the 1149.1 architecture that is used to describe all of the features of an implementation of the 1149.1 hardware inside of a device.

**BSR**: the abbreviation for Boundary-Scan-Register.

**Bypass**: a single-bit register applied in parallel to the BSR that reduces the device's scan shift-path to only 1 bit.

**Capture-Shift-Update Sequence**: three states in the 1149.1 State-Machine involved in a DR-Scan that drives BSR or TDR cells – or involved in an IR-Scan that drives the 1149.1 Instruction-Register – in direct association with

their named actions: Capture allows the cell to observe a signal; Shift allows a cell to pass data from one cell to another cell through a serial shift path; and Update allows a cell to transfer data from the shift path to the parallel output of the cell – this sequence of events is the fundamental basis for how 1149.1 operates.

**DR-Scan**: the use of the *Select-DR Scan* side of the 1149.1 State-Machine – this action is associated with accessing and using a targeted TDR.

**Shift-DR**: a noted sub-operation of DR-Scan that represents the action of shifting data through the defined serial scan-path that is connected between the device's TDI-TDO when the 1149.1-SM is in the *Shift-DR* state.

**IR-Scan**: the use of the *Select-IR Scan* side of the 1149.1 State-Machine – this action is associated with accessing the 1149.1 Instruction Register.

**JTAG**: the acronym for the Joint Test Action Group - the group identity used by the original creators of the 1149.1 standard. JTAG is often used to mean the 1149.1 standard itself rather than the group of originators.

**Public Instructions**: a set of instructions defined in the 1149.1 Standard as being required or optional, but that are fully defined in their operation and the registers they access – the ten public instructions are: **EXTEST**, **SAMPLE**, **PRELOAD**, **BYPASS** (all mandatory), **INTEST, RUNBIST**, **IDCODE**, **USERCODE**, **CLAMP**, and **HIGHZ** (all optional).

**TAP**: the acronym for the Test-Access-Port, the four (optionally five) dedicated 1149.1 pins and associated signal names: **TDI, TDO, TMS, TCK** and, optionally, **TRST***.

**TCK**: an 1149.1 dedicated signal, the Test-Clock that synchronizes all 1149.1 actions.

**TDI**: 1149.1 dedicated signal, the Test-Data-Input that is the beginning of the device's 1149.1 serial scan-path.

**TDI-TDO**: the abbreviation for the 1149.1 serial scan-path inside a device from the device's TDI pin to the device's TDO pin via either the 1149.1-IR or any one of a variety of 1149.1-TDRs.

**TDO**: an 1149.1 dedicated signal, the Test-Data-Output that is the end of the device's 1149.1 serial scan-path.

**TDR**: the abbreviation for the Test-Data-Register – any serial shift-register that operates in accordance with the 1149.1 DR-Scan sequence.

**TLR**: the acronym for the *Test-Logic-Reset* state in the 1149.1 State-Machine – entering this state places all 1149.1 logic into an inert reset state.

**TMS**: an 1149.1 dedicated signal, the Test-Mode-Control that directs the state transitions of the 1149.1 State-Machine on the rising-edge of **TCK**.

**TRST***: an 1149.1 dedicated signal, the active-low asynchronous Test-Reset signal.

**Update-DR**: a noted sub-operation of DR-Scan that represents the action of synchronizing all change actions of the target 1149.1 register or 1687 instrument that is active under the current selected instruction when the 1149.1-SM is in the *Update-DR* state at the falling-edge of **TCK**.

**IEEE 1500-2005**

- the core-test standard that defines core-test-wrappers and the core test-access-mechanism.

**WBY**: the Wrapper-Bypass Register – a single-bit register applied in parallel to the 1500 registers (e.g. WIR, CDR, WDR) that can be used to reduce the Wrapper's scan shift-path to only 1 bit.

**PTDI-PTDO**: the abbreviation for the 1500 parallel port – Parallel-Test-Data-Input to Parallel-Test-Data-Output.

**Select-WIR**: a signal unique to the 1500 TAM that unconditionally selects the Wrapper-Instruction-Register (WIR).

**TAM**: the acronym for Test-Access-Mechanism.

**WBR**: the Wrapper-Boundary Register – a serial shift-register similar to the BSR of a device, but with more defined capabilities, that is meant to wrap around a core to provide the ability to test the core in isolation (in-facing mode), or the test logic attached to the core using the wrapper instead of the core (out-facing mode).

**WIR**: the acronym for Wrapper-Instruction-Register.

**WSI**: a 1500 TAM signal, Wrapper-Serial-Input, that is the beginning of the serial scan-path for the 1500 Test Wrapper.

**WSI-WSO**: the abbreviation for the 1500 serial scan-path, Wrapper-Serial-Input to Wrapper-Serial-Output.

**WSO**: a 1500 TAM signal, Wrapper-Serial-Output, that is the end of the serial scan-path for the 1500 Test Wrapper.

**IEEE P1687**

- the on-chip instrument access and control standard that defines the connectivity and interfaces of embedded instruments other than those associated with 1149.1 and used for board-test.

**Bandwidth**: the use of any data and control mechanisms other than the defined 1149.1 TDI-TDO serial scan-path to deliver/extract more data to/from the instruments:

**PTDI-PTDO**: a Bandwidth concept of using a Parallel set of signals similar to the 1500 Parallel Port and under the synchronization control of the **TCK** – Parallel-Test-Data-Input to Parallel-Test-Data-Output. Also defined in the IEEE 1500 definition section.

**HSTDI-HSTDO**: a Bandwidth concept of using a High-Speed serial set of signals that are synchronized to an alternate and higher-frequency clock than the **TCK** – High-Speed-Test-Data-Input to High-Speed-Test-Data-Output.

**Connectivity-Scheme**: the defined limited number of ways to interconnect 1687 instruments to the top-level 1149.1 logic and interface or to each other:

**Flat**: a Connectivity-Scheme where an instruction in the 1149.1-IR or a local Instrument-IR selects one instrument and connects only that instrument to the TDI-TDO serial scan-path – also called “one-at-a-time”.

**Daisy-Chain**: a Connectivity-Scheme where an instruction in the 1149.1-IR or a local Instrument-IR may select one, many, or all instruments, but all instrument serial scan-paths are active and connected to the TDI-TDO serial scan-path.

**Star**: a Connectivity-Scheme where an instruction in the 1149.1-IR or a local Instrument-IR selects a pre-defined grouping of instruments and only these instruments are connected to the TDI-TDO serial scan-path.

**Concatenate**: a Connectivity-Scheme where an instruction in the 1149.1-IR or a local Instrument-IR selects an instrument and that instrument is added to the already active TDI-TDO serial scan-path – this scheme requires the concept of a 1-hot instruction so that multiple instruments may be enabled simultaneously.

**Hierarchy**: a Connectivity-Scheme where an instruction in a local Instrument-IR of a Parent-Instrument selects a Child-Instrument and enables the selected Child-Instrument to be added into the TDI-TDO serial scan path.

**After**: a Hierarchy Connectivity-Scheme where a Parent-Instrument selects a Child-Instrument and concatenates the TDI-TDO serial scan-path of the Child-Instrument after the Parent-Instrument's TDI-TDO serial scan-path.

**Before**: a Hierarchy Connectivity-Scheme where a Parent-Instrument selects a Child-Instrument and concatenates the TDI-TDO serial scan-path of the Child-Instrument before the Parent-Instrument's TDI-TDO serial scan-path.

**Replace**: a Hierarchy Connectivity-Scheme where a Parent-Instrument selects a Child-Instrument and concatenates the TDI-TDO serial scan-path of the Child-Instrument directly to the TDI-TDO serial scan-path applied to the Parent-Instrument without including the Parent-Instrument's TDI-TSO serial scan-path.

**HIP**: a Hierarchy Connectivity-Scheme acronym for the Hierarchical-Interface-Port – a port on a Parent-Instrument used to pass the Select signal and serial scan-path connections to the Child-Instrument.

**SELi**: a HIP output signal that passes the Select-Instrument signal from a Parent-Instrument to a Child-Instrument.

**WSIo**: a HIP output signal, Wrapper-Scan-Input-output that passes the serial scan-path from the Parent-Instrument to the Child-Instrument.

**WSOi**: a HIP input signal, Wrapper-Scan-Output-input that is the return of the serial scan-path from the Child-Instrument back to the Parent-Instrument.

**WSIo-WSOi**: a HIP abbreviation for the serial scan-path passed between a Parent-Instrument and a Child-Instrument.

**SIB**: a Hierarchy Connectivity-Scheme acronym for the Select-Instrument-Bit – an example is a TDR-Bit that includes a HIP and enables a hierarchical connection.

**ShSIB**: a SIB abbreviation for the Shift portion of the SIB-cell.

**UpSIB**: a SIB abbreviation for the Update portion of the SIB-cell.

**SIR**: a Hierarchy Connectivity-Scheme acronym for the Select-Instrument-Register – an example TDR that is made up of SIBs that each support a HIP and enables multiple hierarchical connections.

**Gateway**: a hierarchical enabled instrument or element whose purpose is to be the only 1687 Shift-Update construct that is allowed to be enabled by an 1149.1-IR Instruction in the 1149.1-Overlap-Zone – and once enabled it becomes an alternate local Instrument-IR that can select other 1687 instruments using only DR-Scans.

**DR-Scan**: (see DR-Scan under the 1149.1 section) – a 1687 Gateway operates through the defined active serial scan-path of 1687 instruments that are connected between the device's TDI-TDO while the 1149.1-SM is in the *Select-DR* state – for 1687 this is how all data and control is delivered to 1687 instruments.

**IR-Scan**: (see IR-Scan under the 1149.1 section) – a 1687 access of the 1149.1-IR to install a "select the Gateway-Enable instruction" (GWEN) when the 1149.1-IR is connected between the device's TDI-TDO and when the 1149.1-SM is in the *Select-IR* state.

**Gateway-Instrument**: an instrument with a Type-B (TDR-like) or Type-C (1500 TAM-like) interface that supports a Hierarchical Connection-Scheme and can be selected by an 1149.1-IR instruction; the hierarchical portion is used as the beginning of the 1687 defined architecture.

**GWEN**: an acronym for Gateway-Enable that represents an instruction placed in the 1149.1-IR that selects a Gateway-Element or Gateway-Instrument.

**Update-DR**: a DR-Scan action of synchronizing the changing of the selection, configuration or use of 1687 instruments when the 1149.1-SM is in the *Update-DR* state and the falling-edge of **TCK** occurs.

**IJTAG**: the synonymous acronym for 1687 – it stands for Internal-JTAG where JTAG is being used as the synonymous acronym for 1149.1.

**Instrument**: any item within a device whose control and configure structure is provided by an 1149.1 TAP and 1149.1 Controller and is optionally accessed by the TDI-TDO serial scan-path. Examples include functional items such as clock-control or bus configuration; Design-for-Test items such as LBIST, MBIST, Mfg-Scan, Embedded Vector-Compression (also known as Test-Data Compression), and 1500 TAMs; Design-for-Debug items such as Trace-Buffers, Embedded Logic-Analyzers, and Bus-Monitors; and Design-for-Yield items such as Ring-Oscillators and Environment-Sensors.

**1687 Instrument**: an instrument that meets the definitions imposed by 1687 – an interface compatible with the ABCD Taxonomy; a connection-scheme limited to the 4 Flat and 3 Hierarchical schemes; and control and configuration driven by an 1149.1 TAP and 1149.1 Controller.

**Hierarchical-Element**: an element is a standalone TDR or TDR-Bit whose only purpose is to provide a hierarchical connection – there is no other functionality that would classify it as an instrument.

**Hierarchical-Instrument**: an instrument that supports a hierarchical connection and can act as a Parent-Instrument to a Child-Instrument – passes on the **Select-Instrument** signal and the WSI-WSO serial scan-path.

**Instrument Interface**: the set of allowed mandatory and optional signals to provide control, configuration, and data sourcing from the defined 1149.1 TAP and TAP-Controller.

**Mfg-Scan**: a single or set of serial scan-paths that are comprised of functional registers, instead of 1149.1 or 1500 control registers, and are generally used in conjunction with ATPG (automatic-test-pattern-generation) – these scan-paths can be concatenated into the TDI-TDO serial scan-path and the clock synchronization can be changed to **TCK** to enable a concept known as scan-dump where the Mfg-Scan Architecture then becomes an instrument to be accessed by 1687.

**Protocol**: the sequence of application of signals needed to operate an instrument. For 1687 instruments, this will require the operation of the 1149.1-SM and the delivery of data through the serial scan-path on the rising-edge of **TCK** when the 1149.1-SM is in the *Shift-DR* state; and actions or configurations will be changed on the falling-edge of **TCK** when the 1149.1-SM is in the *Update-DR* state. These actions can be short-hand referred to as *Shift-DRs* and *Update-DR*s.

**Protocol-Explosion**: the exponential increase in the number of DR-Scans for each level of Hierarchy enabled. This is a result of the Select-Instrument and Select-WIR type signals having to source from one instrument to enable another as opposed to the control being local (self-contained) to the instrument. For example to enable three hierarchical instruments past a Gateway requires: an IR-Scan to select the Gateway; a DR-Scan to select the enable-bit in the Gateway which should select the first hierarchical instrument; a DR-Scan to select the hierarchy-enable bit in the first hierarchical instrument which enables the second instrument; a DR-Scan to select the Select-WIR of the second instrument; a DR-Scan to select the hierarchical-enable bit in the second instrument; a DR-Scan to

de-select the WIR in the second instrument which selects the third instrument; and another DR-Scan to select the WIR in the third instrument.

# Appendix A: 1687 HW Metrics: 10 Evaluation Questions

This document contains the 10 questions, and their associated sub-clarifications, that were used to evaluate the 1687 Hardware Architecture proposals.

## Question #1: concerning Compatibility with Legacy 1149.1 Board-Test Use

**Is the proposed architecture compatible, compliant, and in keeping with the efficiency of the legacy board-level use of 1149.1?**

This question can only be answered **YES** if all of the following considerations are met:

**a.** the architecture will not limit any existing 1149.1 defined operations or instructions;

**b.** the architecture does not require an external filter or logic to condition any TAP signal;

**c.** the architecture will not be adversely affected by random-seeming TDI data that may source from board-level use of the 1149.1 architecture;

**d.** the TAP signals are not required to operate at any clock or data rate above common board-level or in-system capabilities;

**e.** the TDO stream created by the architecture will not adversely affect other 1149.1 compliant devices in the board-level 1149.1 architecture;

**f.** the architecture will not require or result in the 1149.1-SM becoming out of synch with other board or system TAPs;

**g.** the architecture will not require an 1149.1 compliance enable or limit the use of the TAP and TAP controller to certain environments (instruments can be accessed at ATE-test, board-test, in-system, failure analysis, etc.);

**h.** the architecture will not require an 1149.1 compliance enable or special mode or special board-level connections when operated in 1687 mode within the board or system environment (the 1687 architecture does not impact the 1149.1 and vice-versa when in mixed-use – e.g. public 1149.1 instructions and 1687 operations which may be viewed as private 1149.1 instructions can be used simultaneously);

**i.** the architecture will remain compliant to 1149.1 when the board-test architecture is being used (e.g. the TDI-TDO path will not be inverted);

**j.** the architecture will not require a feature that generates a legacy board-level use inefficiency such as an extremely long instruction register.

## Question #2: concerning Standards Compliance, Conflict, and Overlap

**Does the architecture proposal intrude or interfere with any other related standards (ANSI-IEEE, Nexus, Accellera, etc. – aside from 1149.1 handled in question #1)?**

This question can only be answered **NO** if all of the following considerations are met:

**a.** the architecture does not require non-conformance or non-compliance of any portion of a related standard;

**b.** the architecture does not re-define an architecture description of an existing standard;

**c.** does not "obsolete" a portion of an existing standard.

## Question #3: concerning Instrument Use and Reuse

**Does the architecture support access to a wide variety of new and existing instruments with a minimum amount of interface adjustment, engineering development, and design cost?**

This question can only be answered **YES** if all of the following considerations are met – the hardware architecture proposal:

**a.** does not exclude any existing common instruments or instrument interface types;

**b.** does not incur a significant engineering penalty to interface to 1149.1 compatible instruments;

**c.** supports sequences, methods, or operations that enable use of instruments that would be limited if restricted to the current traditional 1149.1 operations;

**d.** supports the concept of System-on-Chip re-use where a device supporting the proposed architecture becomes an embedded core within a larger device;

**e.** supports the concept of "compliant" instruments or instrument interfaces;

**f.** supports the concept of plug-and-play for compliant instruments.

## Question #4: concerning Hardware Architecture Adoptability

**Does the architecture proposal contain any absolute or non-resolvable barriers to adoption?**

This question can only be answered **NO** if all of the following considerations are met – the hardware architecture proposal:

**a.** the architecture proposal does not violate the IEEE patent policy;

**b.** does not contain non-implementable technology;

**c.** does not represent a vendor-specific solution that will not be supported by competing vendors;

**d.** does not intrude or conflict with existing standards.

## Question #5: concerning Growth and Longevity of the Standard

**Is the architecture proposal flexible enough to grow with Moore's Law so it can have a reasonably long lifetime?**

This question can only be answered **YES** if the following considerations can be met – the architecture proposal is still valid and efficient when:

**a.** the instrument count grows into the thousands, tens-of-thousands, or more;

**b.** the device's internal and/or external interconnect data-rates grow to become Giga-Bit only busses;

**c.** the instrument signal interfaces grow to become extremely wide control/status/IO busses;

**d.** the device's data and control bandwidth requirements grow by orders of magnitude;

**e.** more on-chip intelligence is required to reduce operation, application, or safety latency;

**f.** extreme power management must be applied to on-chip structures – possibly including the instruments.

## Question #6: concerning Engineering Impact

**Does the architecture support design and implementation flexibility against engineering goals and concerns?**

This question can only be answered **YES** if all of the following considerations can be met:

**a.** a system of adjustments or trade-offs exist between area, power, timing, gate-count, routing, loading against critical paths, timing-closure, schedule, test-time/cost, etc. (for example, instrument grouping options, routing width options, etc.);

**b.** the architecture supports definable and reasonable goals for area impact, timing impact, power consumption, routing width or routing congestion impact, signal and interface timing targets, clock skew targets, etc.;

**c.** the architecture insertion and optimization is possible under the control of existing design and implementation tools (for example, synthesis constraints can be used to drive some trade-off decisions);

**d.** the architecture does not require fixed and inflexible architectural features (for example, custom or hard-macro layout units).

## Question #7: concerning Engineering Design Flow Compatibility

**Does the architecture support human understanding, machine automation and compatibility with modern design flows?**

This question can only be answered **YES** if all of the following considerations can be met:

**a.** the architecture is describable in human understandable forms such as state diagrams, timing diagrams, bus transfers and transactions, logic diagrams, etc.;

**b.** the architecture is describable in modern modeling and netlist languages (.e.g. HDL, RTL, Gates – Verilog and VHDL);

**c.** the architecture's requirements and operations are describable in terms of design constraints (synthesis, timing analysis, layout placement, routing restrictions, etc.).

## Question #8: concerning Instrument Safety

**Does the architecture support a default quiescent configuration and non-disruptive and safe features?**

This question can only be answered **YES** if all of the following considerations are supported – the architecture supports:

**a.** reset, idle, and/or quiescent non-disruptive default states that use minimal power;

**b.** a pre-conditioning or setup operation to prepare the instrument for operation if needed;

**c.** a non-disruptive query for instrument status if status is provided by the instrument;

**d.** a recovery operation to recover from illegal states or non-safe operation (such as an error status interrupt and a reset-assertion response).

## Question #9: concerning Instrument Organization and Communication

**Does the architecture support instrument organization, scheduling, and communication?**

This question can only be answered **YES** if all of the following considerations are supported – the architecture supports:

**a.** some form of instrument grouping or organization (such as flat, hierarchical, daisy-chain, star, concatenation, etc.);

**b.** some form of individual instrument communication;

**c.** some form of simultaneous instrument operation with some instruments or groups of instruments;

**d.** some form of broadcast or simultaneous instrument communication (such as a trigger or update) that can synchronize the activity or operation of multiple instruments;

**e.** some form of instrument-to-instrument communication for some instruments or instrument groupings (parent-to-child, master-to-slave, peer-to-peer, etc.).

## Question #10: concerning Architecture Bridge to Languages

**Is the interface structured and simple enough to allow it and its operation to be described in a description and protocol language that will meet the language metric criteria?**

This question can only be answered **YES** if all of the following considerations are met:

**a.** there is a finite and reasonable set of defined interface signals (e.g. not an open policy of allowing any signal as long as it is described);

**b.** there is a finite and reasonably-sized defined protocol or set of sequences that can be applied to the interface signals to operate them correctly and safely (e.g. not an open policy of allowing any set of sequences as long as they are described);

**c.** that any safety and/or recovery operations associated with the interface are easily-understandable and describable;

**d.** that any illegal or non-supported signal states or sequences can be described clearly;

**e.** that the evaluation of the languages involved with the interface meets the criteria (correct expected answer) of the Language Metric questionnaire.

# Appendix B: 1687 Instrument Interface Conversions

This appendix presents “Example and Clarification” material focused on dealing with instrument interfaces that are not operable by a compliant 1149.1 Controller; or instruments that do not support hierarchy.

## Instrument Conversions

**The 1687 proposed hardware architecture allows for the conversion of one type of 1687 instrument interface to another type (to support hierarchy, different connectivity, or to better meet an engineering tradeoff); and allows for the conversion of non-compatible-to-1687 instrument interfaces to one of the defined 1687 interface types.**

Conversions are preferred and allowed for several reasons. The main reason is because it is preferable not to support instruments that are not controlled and configured by the 1687 system (although, it is allowed for instruments to exist and not be part of 1687) – since this would imply an alternate access system must exist and be supported. Another reason to allow conversions has to do with the current state of instrument re-use and legacy instruments – it is not expected that current existing instruments will have 1687-compliant interfaces or will support hierarchy as needed. It is expected that most instruments today are going to be Flat, Star, or Daisy-Chain connections directly to the 1149.1 TAP-Controller – which is not scalable when the instrument count increases into the high-hundreds to thousands of instruments.

The most common expected interface conversion is to add hierarchy to an instrument that does not support hierarchy. This can be done to any 1687 instrument type to enable it to be used to control other instruments – or to enable any instrument type, except a Type-D, to be used as a Gateway.

The second most common expected interface conversion is the use of a legal 1500 TAM that supports the Transfer Function (a non-1149.1-compatible sequence); or support the use of an alternate-clock (not **TCK**); or support an alternate control-data-port (not the TDI-TDO serial scan-path).

Another expected interface conversion will be the use of multiple 1149.1 Controllers on one device – although there are many ways to accomplish this, having them all be of equal priority is a violation of 1149.1, and making all but one subservient is not done in a standardized manner. This largely occurs because re-use IP-Cores are delivered with intact TAPs and 1149.1 Controllers.

Examples of these conversions will be presented and for these examples, SIBs will be used as the method to implement the conversion.

**Simple Instrument Conversion**

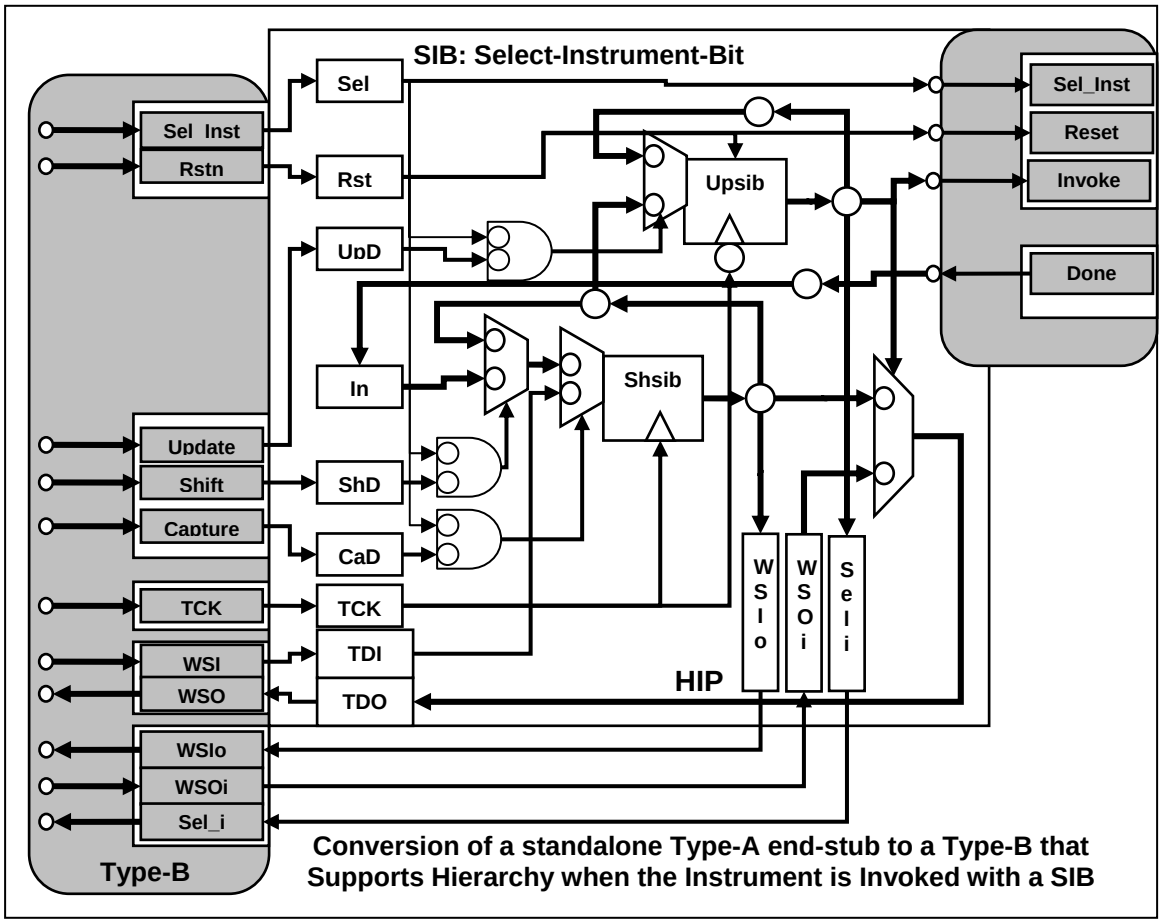


**Figure B-1: An example Conversion of Type-A to Type-B+Hierarchy**

Figure B-1 shows a Type-A instrument that may represent a Logic-BIST or Memory-BIST that requires only an Invoke and a Reset after it is Selected – and produces only a Done signal after it has completed its self-contained process (and the Done not transitioning after a number of cycles or an amount of time represents a fail-condition). Normally, this type of instrument would have its signals brought directly to the device's pins, but in the case of supporting hundreds of these as on-chip instruments and the need to select only a few at a time due to power consumption restrictions – there may not be enough device pins to provide any type of reasonable and adjustable "test-scheduling".

So, in this case the addition of an 1149.1-compatible TDR-like interface that adds the instrument to the TDI-TDO serial shift-path; provides an Invoke signal that is synchronized to the *Update-DR* event; and can capture the static Done signal through a *Capture-DR* event – can allow the Type-A instrument to be controlled by the 1149.1 TAP-Controller through the 1687 architecture. In addition, when this instrument is "invoked" and active (when the UpSIB has a logic 1), the Hierarchical-Interface-Port (HIP) is enabled. The HIP can be connected to some

other instrument (which can become active from the Sel_i signal, or may require a DR-Scan and an *Update-DR* event to become active); or the HIP can be connected to another SIB to enable further hierarchical instrument connections.

**Complex Instrument Conversion**

In some cases, an instrument needs a complex transformation that includes a state-machine or other complex sequential conversion. One common example of this is the legally supported Transfer-Sequence identified in the IEEE 1500 Core-Test Standard. This sequence is required as a result of re-using a functional register as both the Capture and the Update register. From the 1149.1 operation point of view, the functional register should conduct a capture of the values on the core input pins in the *Capture-DR* state and the very next state is the *Shift-DR* state which should bring the value to an observe point by shifting it through the TDI-TDO serial scan-path. However, the functional register that captured the core input pin value is not part of the core-wrapper scan chain – so, the needed sequence is to "capture" core input values into the functional register, "transfer" the captured value to a TDI-TDO serial scan-path register, and then "shift" the value to an observation point.

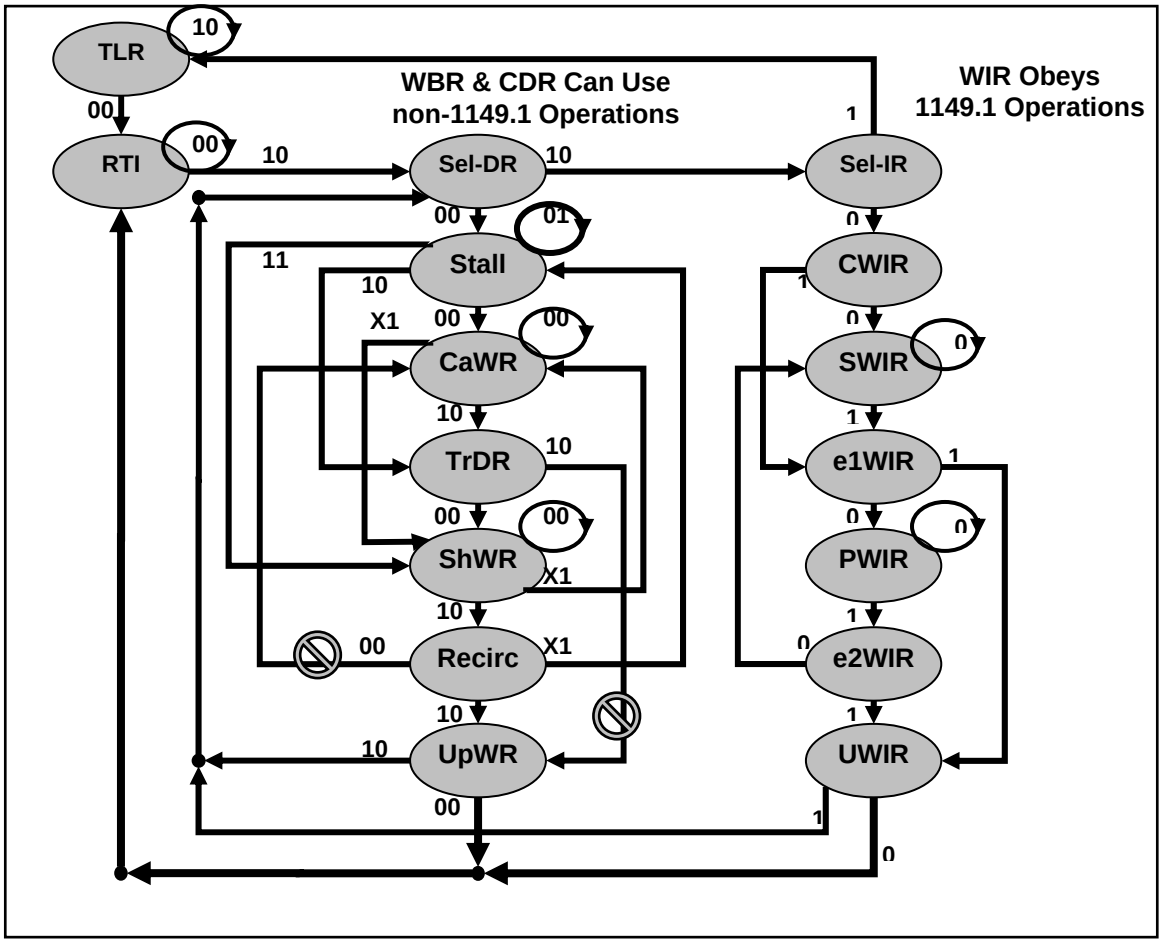


**Figure B-2: An example Complex 1500-TAM+Transfer State-Machine**

Figure B-2 shows an example of a state-machine similar to the 1149.1-SM, except the states and transitions involved with the Data-Register-Side have be re-mapped to provide the necessary sequences for the 1500 Core-Test Wrapper that supports Transfer. As can be seen, there is a “Transfer Data-Register” function (TrDR) between the Capture-Wrapper (CaWR) and the Shift-Wrapper (ShWR) states. In addition, the CaWR state now loops so multiple capture cycles can be supported.

The Instruction-Side of the state-machine is exactly the same as the definition for the 1149.1-SM since it will be used to create the sequences used by the WIR – and the WIR does not include any odd sequences such as “Transfer”. Note that this state machine requires two mode signals to conduct the transitions – a TMS and a TMS-2. If TMS is operated, and TMS-2 is held to logic 0, then the state machine operates exactly like the 1149.1-SM – only TMS-2 enables the additional operations.

Given that there is now a state-machine that can operate the 1500 Core-Test Wrapper with Transfer, how can it be applied without violating the 1149.1 requirements to not support multiple-TAPs or to change the nature and function of the defined system-level Master-TAP? One way is to couple the state-machine and the 1500 Core-Test Wrapper and to treat them as one instrument – thereby converting the Type-D 1500 Core-Test Wrapper (Type-D because it supports the Transfer function) into a Type-B or Type-C instrument.

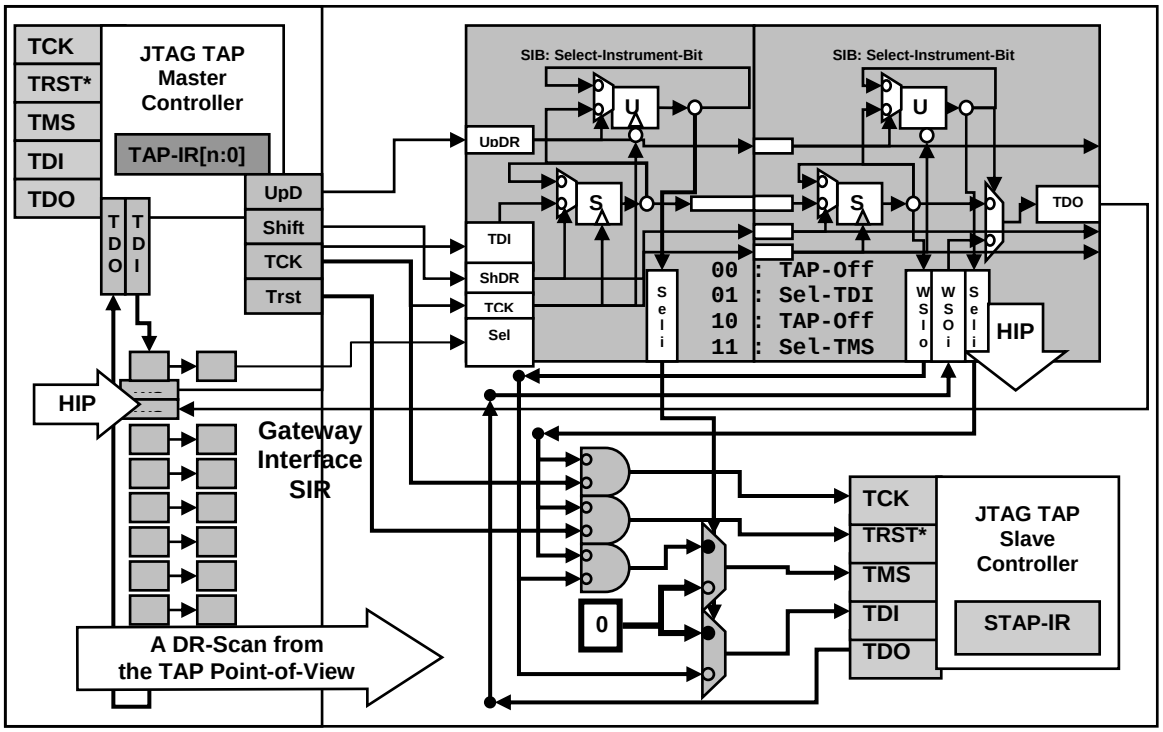


**Figure B-3: An Example State-Machine Instrument Conversion**

From the 1687 point of view, the Slave-TAP is not a valid instrument because it requires a **TMS** signal – and that is not allowed by the current specification. However, by placing a SIB and a Slave-TAP swap circuit (STAP-SWAP) circuit in front of the Slave-TAP, the Slave-TAP is converted into a Type-B instrument that begins at the SIB interface (Select-Instrument, **TCK**, ShiftEn, UpdateEn, TDI, and TDO). Figure B-3 shows an example of a 2-Bit SIB connected to a Gateway as a Type-B instrument and driving a STAP-SWAP circuit. Note, to support more than one **TMS** signal – extra SIBs can be used to create more **Select**-like signals.

The STAP-SWAP converts the Master-TAP **TDI** data-stream into both the TMS-data and the TDI-data for the Slave-TAP. This allows the Slave-TAP to be operated with only a DR-Scan from the Master-TAP by shifting TDI-data into the SIB and conducting *Update-DR* to select either the TMS-path or the TDI-path – and then passing the Master-TAP TDI-data on to the Slave-TAP. This makes use of the fact that the TDI-path for the Slave-TAP is only valid when the Slave-TAP is in the *Shift-DR* state and so the Slave-TAP TMS input pin must remain at a logic 0 (to keep the *Shift-DR* state looping on itself) during this operation. When the TMS is being actively operated and the Slave-TAP is not in the *Shift-DR* state, the data on the Slave-TAP's TDI is not relevant.

The Slave-TAP can be a modified TAP-Controller such as the one shown in Figure B-2, and it can now be connected to a compliant 1500-TAM interface that supports the Transfer-Function. The Slave-TAP's Instruction-Register can be used to select the 1500-TAM, and the ShiftEn, CaptureEn, UpdateEn, and TransferEn signals can be applied directly.

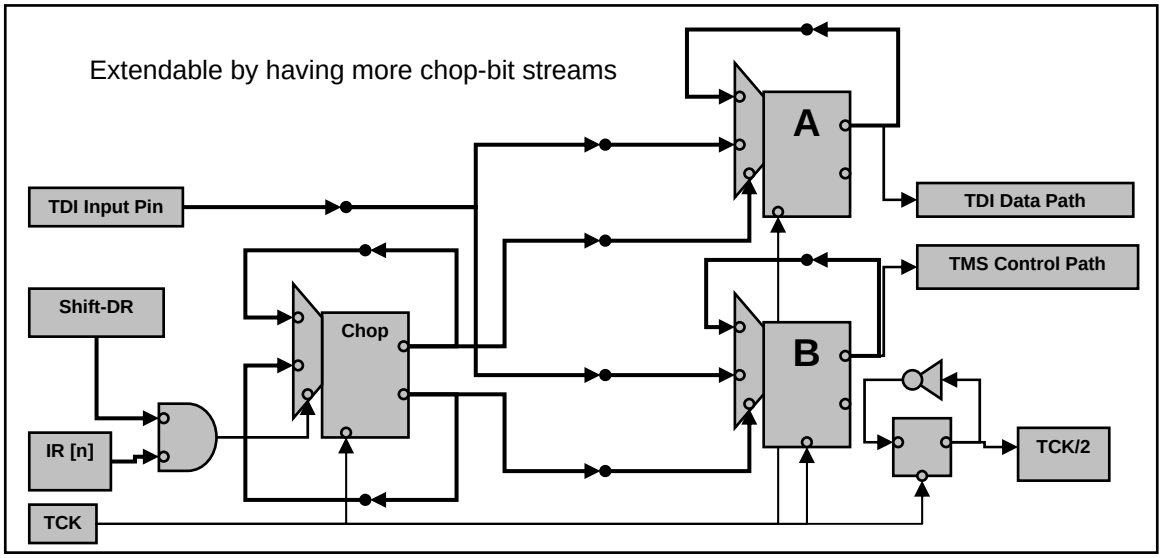


**Figure B-4: An example Sequential TDI-Data Chopper**

The STAP-SWAP is not the only solution to this problem. Figure B-4 shows a sequential version of the STAP-SWAP known as a TDI-Chopper where the

Master-TAP-level TDI data-stream is automatically separated into the Slave-TAP's TDI-data and TMS-data. This effectively divides the data by two, so to keep data-synchronization with the sequential actions of the Slave-TAP, the clock applied to the Slave-TAP and its instruments should be **TCK** divided-by-two. Note that the IR[n] input can be delivered from a 1500-WIR or from a SIB's HIP Sel_i signal.

**Localizing an Instrument**

The SIB can be used to (convert) make an instrument "self-contained" in that its Select and Select-WIR signals can be supplied locally. The reason to do this conversion is to minimize the "Protocol Explosion" that happens when hierarchical depth is applied to instrument connections.

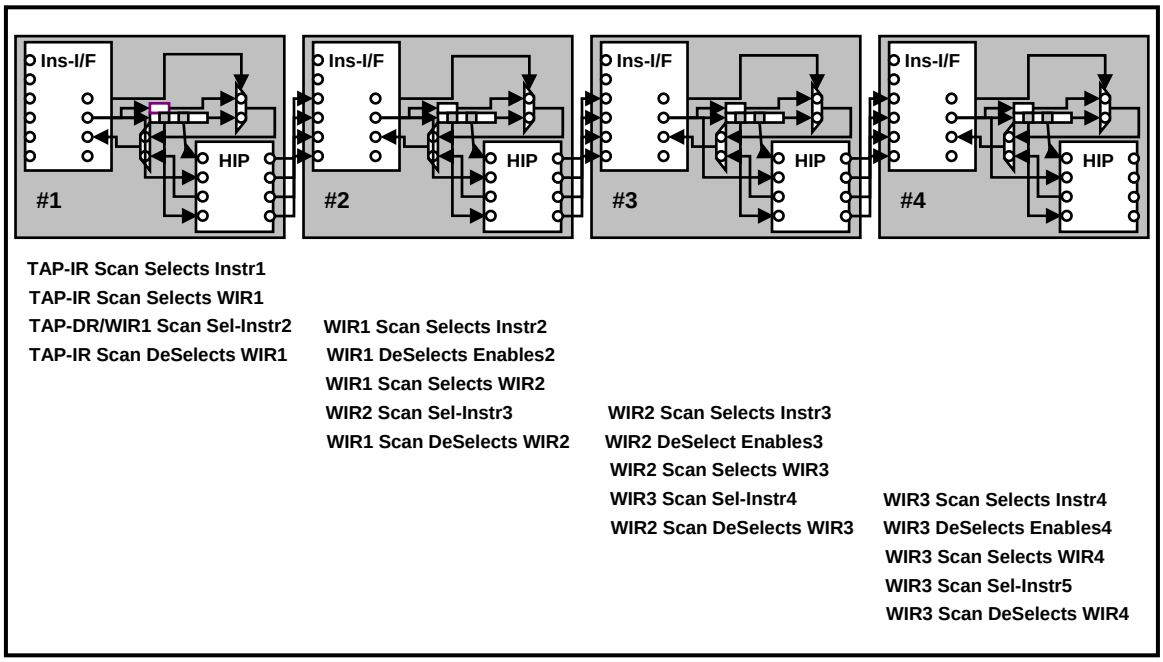


**Figure B-5: An example Hierarchical-Access Protocol-Explosion**

As can be seen in Figure B-5, each access to the next instrument in line requires the operation of the previous instrument – known as the Parent-to-Child interaction. A Parent must be selected, then the Parent must be operated by having different instructions activated – and one or more of those instructions will select the next instrument, the Child. Installing different instructions into the Child may require operation of the Parent to install different instructions. This leads to an ever growing trail of operations, or Protocol-Explosion, the further out into the hierarchy the control needs to propagate.

The way to reduce this Protocol-Explosion is to require each instrument to only have to enable its HIP and to make each instrument itself contain its Select-instrument and Select-WIR control. This can be done by associating a SIB with each instrument.

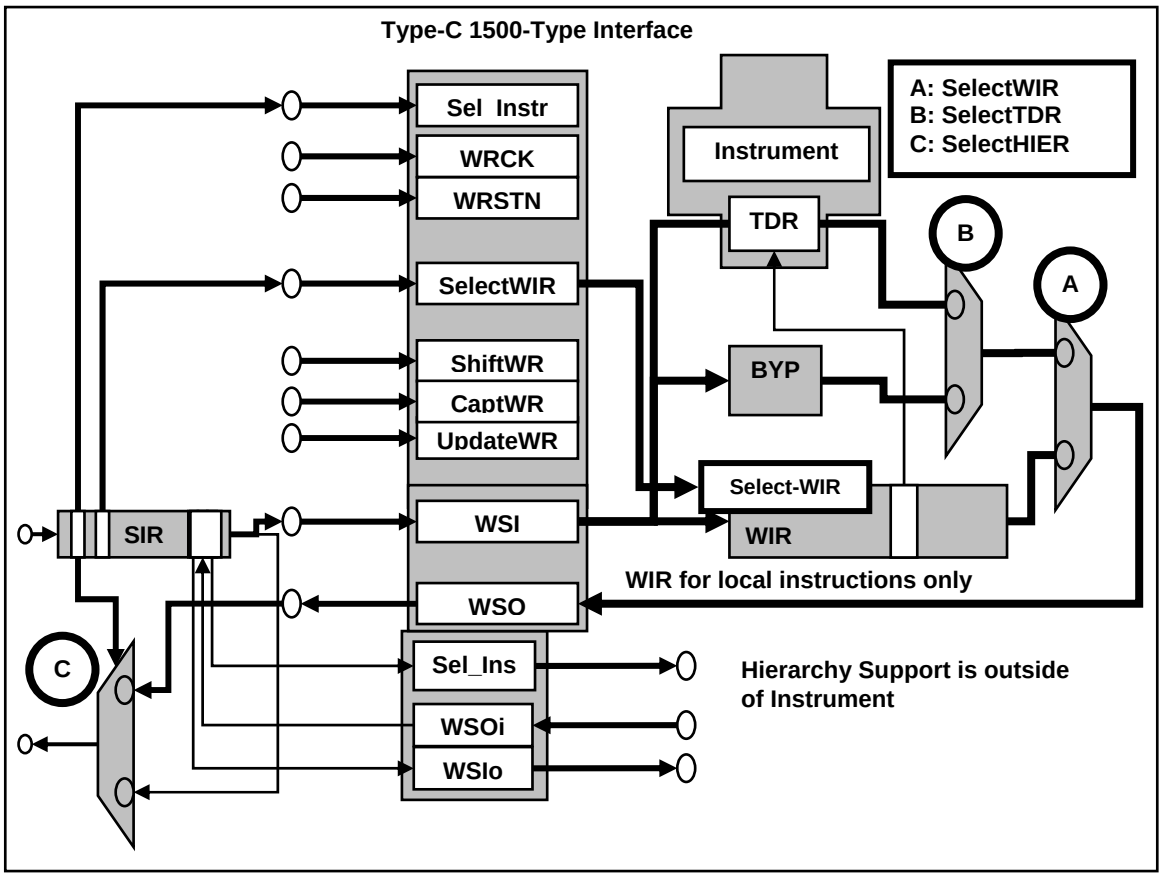


**Figure B-6: An example of an Instrument with Localized-Control**

Figure B-6 shows an example of a 1500-TAM that has localized **Select** and **Select-WIR** control added in-front and in-line with the instrument, and additionally, has a HIP added externally by using a 2-Bit SIB contained within a SIR. This conversion changes an ordinary 1500-TAM into a self-contained re-usable hierarchical-instrument that eliminates much of the Protocol-Explosion since a single DR-Scan through the in-line SIR can enable the instrument and select the WIR for operations.

# Appendix C: 1687 Instrument Connectivity Tradeoffs

This document presents “Example and Clarification” material on exploring and, implementing the different connectivity-schemes versus the needs and goals of the end-user architecture and the cost-factors involved with the engineering-tradeoffs.

## The Connectivity-Schemes and Tradeoffs

**The 1687 proposed hardware architecture allows for multiple different connection schemes – 4 non-hierarchical configurations and 3 hierarchical configurations. Each configuration has pros and cons associated and has more or less efficiency depending on the size of the device and the number of instruments that need to be supported. The allowances are so that instrument connections can fit within schemes driven by engineering, economic, efficiency, and utility tradeoffs.**

The connectivity and communication schemes applied to any given instrument architecture are very important to meet the end goals of the instruments within a device. In some cases, the instruments that are to be controlled are functional items such as clock-controllers, power-controllers, and bus-configuration-controllers that may naturally be used when the instrument is in its final packaged implementation; in other cases, the instruments that are to be controlled are test-only or debug support items that may be targeted for use in just one environment (such as ATE testing) or may not be used at all unless the device does not work as predicted. Considering each use environment for each grouping of instruments results in different items being important:

1. For functional instruments, the important factors may be ease-of-access, latency-of-application, and allowance of alternate-control – the consideration of how long in time or in clock-cycles it takes to access the instrument and for the instrument to then become active after some directive, condition or event occurs; and the ability for a functional master to control the instrument as well as the 1149.1 TAP-driven 1687 system.

2. For test instruments, the important factor may be “test-scheduling” and “flexibility” – the ability to select and choose which instruments can operate simultaneously to reduce test time, but to be able to change the concurrence-grouping of instruments if another issue such as power-consumption or thermal-overload becomes an issue.

3. For debug instruments, the important factor may be to “not impact functional operation”, flexibility-of-use, and bandwidth – the ability to access and use any instrument at any time to install or monitor a condition, or to collect data and to then be able to efficiently transport that data to an analysis point (inside the device or outside of the device).

Because all of these cases (and more) may be true within one device; and because the goals or needs may change depending on the size of the device or the number of instruments to be integrated – there is no simple connection-scheme that can meet all needs. Similarly, no single "fixed" connection architecture description can meet the needs for all devices since the size, the number of gates, the mix of memory versus logic, the cost, the number of instruments, the frequency performance, the power-consumption needs, etc., can change radically from market-to-market and from device-to-device.

The main body of the 1687 Hardware Proposal describes the connection options and the tradeoffs that need to be considered – they are summarized here in table C-1. The assessment uses terms such as None or Minimal, Low, Medium, High, and Excessive or Maximum – these are relative terms in that there exist configurations with given numbers of instruments where one connectivity-scheme may have an advantage over another scheme, but if the instrument count were allowed to grow into the thousands, then that scheme would get the description listed in table C-1 and based on the most-pessimistic perceptions.

| **Scheme** | **Flat** | **Daisy-Chain** | **Star** | **Concatenate** | **Hierarchy** |
|---|---|---|---|---|---|
| Tradeoffs | | | | | |
| **Engineering**: configuration impacts that affect the device's engineering budgets | | | | | |
| Area/Logic | Medium | Minimal | Low | High | High |
| Routing | Excessive | Low | Medium | Low | Minimal |
| Power | Minimal | Excessive | Medium | Minimal | Minimal |
| Risk | Minimal | Excessive | Medium | Medium | Low |
| **Efficiency**: configuration impacts that affect the use/compliance of 1149.1 | | | | | |
| Scan-Path-Depth | Minimal | Excessive | Medium | Minimal-to-Excessive | Minimal-to-Excessive |
| Scan-Path-Stability | Stable | Stable | Stable | Dynamic | Dynamic |
| IR-Depth | Excessive | Minimal | Medium | Low | Minimal |
| **Utility**: configuration impacts that affect the automation and usefulness of 1687 | | | | | |
| Language-Complexity | Minimal | Minimal | Medium | High | Excessive |
| Protocol-Complexity | Minimal | Minimal | Medium | High | Excessive |
| Concurrence | None | Maximum | Medium | Maximum | Maximum |
| Flexibility | None | None | Medium | High | Maximum |
| Re-use | Minimal | High | Medium | High | Maximum |

**Table C-1: Connectivity versus the Number of Instruments**

Figure C-1 shows an example of one interpretation of the efficiency of the various connectivity schemes versus the tradeoffs as the number of individual addressable instruments grows in number. Note that this interpretation of

connectivity-schemes is an assessment of “one connectivity-scheme versus the others” and does not address mixed-connectivity-schemes:

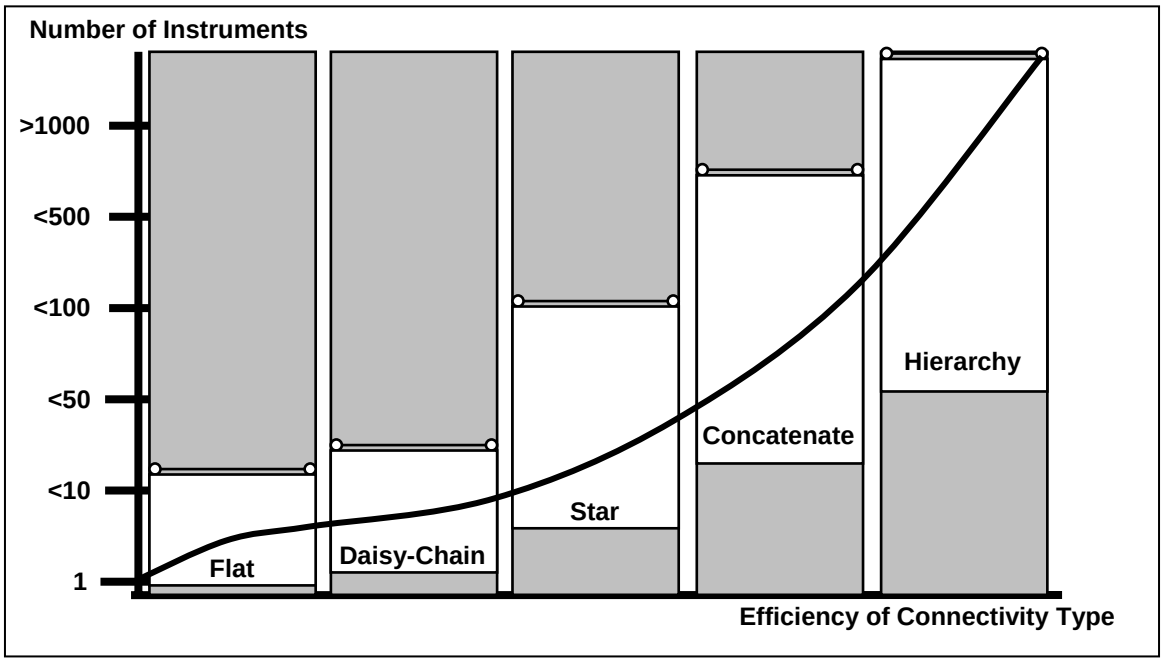


**Figure C-1: Connectivity versus the Number of Instruments**

**The Flat One-at-a-Time Connectivity-Scheme**
The Flat connectivity-scheme has a high routing cost in that there are individual connections that are activated by individual mutual-exclusive instructions – for this reason it is best applied when there are very few instrument interfaces to address (of course, other factors apply such as the size of the device or the amount of logic supported as compared to the number of instruments). As the number of instrument interfaces that need to be connected into the access architecture grows, the lack of concurrence (parallel instrument operation), lack of test-scheduling flexibility, and the dominance of the routing-congestion impact will make this scheme less effective.

**The Daisy-Chain All-at-Once Connectivity-Scheme**
The Daisy-Chain connectivity-scheme has a high risk factor and high power-consumption cost in that all instruments have their serial scan-paths active whenever a single instrument needs to be active and a break anywhere in the single scan-path denies access to all instruments – for this reason it is best applied when the number of instrument interfaces to integrate are just a few (for example, >1 but less than 50). However, the Daisy-Chain should be used instead of the Flat if there is an instrument concurrence requirement.

**The Star Groups-at-Once Connectivity-Scheme**
The Star connectivity-scheme is a mixture of Flat and Daisy-Chain in that it is the organizing of several instruments into several different Daisy-Chain groupings

and then providing Flat access to one group at a time with mutual-exclusive instructions. This scheme offsets the routing-congestion impact, the risk-factors, and the power-consumption impact, but only for a limited amount of growth.

At some level of instrument growth, these factors will begin to dominate again (fewer than a hundred shown in this interpretation). The most-likely loss of efficiency will occur in order to support test-scheduling and instrument operation concurrence – the number of instruments that need to be included into Daisy-Chain groups will grow to the point that risk and power will begin to dominate again and the amount of post-silicon flexibility (the ability to adjust the number or grouping of active instruments once in silicon) will drop. Any concurrence and flexibility will have to be planned ahead of time and placed into the architecture during the design (pre-silicon) phase.

**The Concatenate Add-When-Active Connectivity-Scheme**

The Concatenate connectivity-scheme will mediate the risk-factors and the power-consumption factors associated with Daisy-Chain connections, and will return the concurrence and flexibility needs of supporting a wealth of instruments to the Flat connections. The routing-congestion can be managed by providing all of the control signals to all of the instruments on a bus, and providing only a single activation select-line per instrument. The bypassing of any non-active instruments should include a local Bypass Register, so at some point, the number of in-line Bypass Registers may become a power or shift-efficiency problem (the connectivity is still very much like a Daisy-Chain, but with only the Bypass Register being active for each bypassed instrument). For this reason, the efficiency of the architecture will reach a limit at some number of supported instruments (shown in this interpretation as being near 500).

**The Hierarchical Add-When-Active Connectivity-Schemes**

The Hierarchical connectivity-schemes are required to provide an efficient and flexible test-scheduling architecture as the growth of the number of instruments becomes very large. Even though the hierarchical method results in the most flexibility, least power-consumption, least routing-congestion impact, and the easiest test-scheduling capability – the hierarchical method complicates the description language and the access protocol. However, when the device supports over 1000 instruments, then the protocol and access mechanisms will be complex in a different way. The other reason to use a Hierarchical connectivity-scheme, when the instrument count isn't as high, is the natural support of being able to hide the non-active instruments completely for IP protection.