A MODEST PROPOSAL FOR CHIP-LEVEL CONSOLIDATION OF A MULTIPLICITY OF EMBEDDED TAPS

(WITH CONSIDERATION FOR

JTAG BOUNDARY SCAN, JTAG SOFTWARE DEBUG, AND

IJTAG INSTRUMENTATION NETWORKS)



REVISION A | SPECIFICATION

BY AL CROUCH, ADAM LEY, LARRY TRAYLOR







Al Crouch – Chief Technologist, Embedded Instrumentation Methodologies and IJTAG

Al investigates the use of embedded instruments for IC test and debug, board test and debug, and software debug. He is a Senior Member of the IEEE and served as the vice chairman of the working group that developed the IEEE 1687 IJTAG standard for embedded instruments. He contributed significantly to its hardware architecture definition. Al is also a member of the working group developing the

IEEE P1838 standard for 3D die stack test and debug. Al's previous experience includes designfor-test and debug engineering at various semiconductor companies, including Texas Instruments, Digital Equipment Corp. and Motorola, as well as chief scientist at startup companies DAFCA and INOVYS.



Adam Ley – Chief Technologist, Non-intrusive Board Test and JTAG

Adam ensures that ASSET's non-intrusive board test (NBT) methodologies meet the increasing need for improved board test coverage in the face of disappearing physical access. In keeping with ASSET's strong commitment to industry standards, Adam is an active participant on the IEEE 1149.1 boundary-scan/JTAG working group,

having served terms as vice chair and as technical editor (for the 2001 revision), as well as in nearly all related standards, including: 1149.4, 1149.5, 1149.6, 1149.7, 1149.8.1, 1500, 1532, 1581, P1149.1.1, P1149.10, iNEMI boundary-scan adoption, PICMG MicroTCA and SJTAG (system JTAG). Before ASSET, he had roles for over a decade at Texas Instruments, Sherman TX, in application support for boundary-scan logic products and in test and characterization of new logic families. Adam earned a BSEE degree from Oklahoma State University, Stillwater OK.







Larry Traylor, Vice President Software Debug and Trace

Larry is a recognized technology leader in the area of both Intel® and ARM® debug and trace architectures having worked on a continual basis with debug architecture teams of both processor companies. Larry has also consulted with numerous ARM licensees on CoreSightTM implementations for specific SoCs to optimize the value for the software engineer to achieve his job goals in debugging complex embedded

software applications. Prior to joining ASSET InterTech, Larry co-founded Arium Corporation and served as the company's president, CEO and chairman of the board. He was instrumental in driving Arium's vision of hardware-assisted software debug and code trace tools. Larry joined ASSET in 2013 when Arium was acquired. He has a BSEE from Cal-Poly Pomona.





Table of Contents

Executive Summary	6
Introduction	7
Problem	7
Embedded TAP (eTAP) Defined	8
Requirements	9
JTAG Boundary Scan (IEEE 1149.1-including 1149.4, 1149.6, 1149.8.1-and/or 1149.7)	9
IEEE 1149.1	10
IEEE 1149.7	12
JTAG Software Debug (ARM DAP, Intel ITP and others)	13
Conformance to JTAG BYPASS	13
Compatibility to Automated Discovery	13
IJTAG Instrumentation Networks (IEEE 1687 including IEEE 1500)	15
TDR-Wrapped Instruments	15
eTAP-Connected Instruments	16
Proposal	17
Rules	20
Conclusions	26
We Can Help!	26

© 2015 ASSET InterTech, Inc.

ASSET and ScanWorks are registered trademarks and the ScanWorks logo is a trademark of ASSET InterTech, Inc. All other trade and service marks are the properties of their respective owners.





Table of Figures

Figure 1: Conventional test logic architecture defined by IEEE 1149.1	. 10
Figure 2: TAP controller state diagram	. 12
Figure 3: Proposed eTAP selection network (one slice)	. 18
Figure 4: Proposed chip-level architecture for multiple eTAPs that are always available (TDO side)) . 19
Figure 5: Alternative chip-level architecture for multiple eTAPs with instruction-limited availability (TDO side)	. 20





Executive Summary

Typically, the development of complex systems-on-chip (SoC) or systems-in-package (SIP) involves integrating multiple intellectual property (IP) blocks, each with its own embedded instruments for characterization, verification, validation, test, software debug or some other application. For many of these embeddable IP blocks, the instruments therein are accessed through a Test Access Port (TAP) as defined by IEEE 1149.1 (commonly known as JTAG).

When it was developed in the 1980s, JTAG specified the TAP as a port on a chip. Yet, practices for inclusion of the TAP as a port on IP blocks, which are designed for integration into SoCs and other types of devices, have been largely undefined. In fact, without due consideration and planning during the chip design phase, the integration of multiple embedded TAPs (eTAP) into a single device may not yield the desired chip-level capabilities. For example, conflicts may arise among embedded instruments across various application domains such that their operations are compromised or coordination among embedded instruments may be impeded. In a worst case, access to the required embedded instrumentation from a given tool chain, such as software debug tools, might not be possible at all. Furthermore, device compliance to certain standards, including IEEE 1149.1, IEEE 1149.7, IEEE 1500 embedded core test, IEEE 1687 Internal JTAG (IJTAG) and others, could be jeopardized.

This specification offers chip designers a set of guidelines by way of recommended practice, which, if followed, will avoid these problems, allowing both chip and system designers to benefit from the full functionality embodied in the complete array of embedded instrumentation within all of the IP blocks on a complex chip.





Introduction

Integrated circuit (IC) design practice is trending strongly toward system-on-chip (SoC) and system-in-package (SiP) methodologies, wherein complex intellectual property (IP) blocks, whether implemented as hard or soft cores or as known-good die (KGD), are integrated to create system solutions on a single substrate and/or in a single package. Fortunately, many, if not most, of these IP blocks will be fitted with their own suites of embedded instruments for application domains such as test, debug, validation, configuration, tuning or others.

In addition, much of this instrumentation across the various application domains employs the JTAG Test Access Port (TAP) as the main point of access to individual instruments. Yet the very ubiquity of the TAP for such embedded applications gives rise to questions and concerns because the TAP, while well-defined by the IEEE 1149.1 JTAG standard for chip level access, is less well-defined and understood for embedded use.

Problem

Historically, the instruments for various application domains and modes were placed in different architectural groupings, each accessed through different mechanisms. However, today's reliance on electronic design automation (EDA) tools to create, insert, verify and synthesize SoC and SiP designs has consolidated all of the different access mechanisms under the JTAG TAP.

Of course, different hardware interfaces and tool chains still support the various embedded instrument applications, including automated test equipment for structural test, low-cost scan kits for hardware debug and run-control emulation probes for software debug. If each of these application environments is to operate as expected, the various embedded instruments must be accessible within the overall test and debug architecture of a given design, especially if instrument interference is to be avoided and instrument coordination in some fashion enabled.

But, given the recent design trends involving a TAP on each instrument and/or IP block, multiple embedded TAPs (eTAP) can rapidly proliferate behind the primary, chip-level TAP (CLTAP). Consequently, some guidelines must be observed to ensure that this complex access mechanism does not break down, break the tool chains or jeopardize compliance to the pertinent standards, such as IEEE 1149.1, IEEE 1149.7, IEEE 1500 and IEEE 1687.





Embedded TAP (eTAP) Defined

An eTAP is a TAP and its TAP controller, like those defined by IEEE 1149.1 JTAG, but embedded in IP in a complex chip so that it functions as a child in relationship to the parent CLTAP and its chip-level TAP controller (CLTAPC).

Every eTAP, following the model of the IEEE 1149.1 JTAG standard, is comprised of the following:

- signals: eTCK, eTDI, eTDO, eTMS and (optional) eTRST* (like the TAP signals in Figure 1)
- controller: 16-state finite-state machine (eFSM) (like the state diagram of Figure 2)
- registers: instruction register (eIR) and test data registers (eDR).

Although the IEEE 1149.7 standard designates an embedded TAP controller differently, as EMTAPC, its definition, consistent with the discussion above, is given as:

""EMTAPC" is a TAPC other than the CLTAPC implemented within the System Test Logic."

IEEE Std 1149.7-2009 - Reprinted with permission from IEEE. Copyright IEEE 2010. All rights reserved.

*NOTE: In limiting the scope of the EMTAPC to the System Test Logic, IEEE 1149.7 makes allowance for its own adapter TAP controller (ADTAPC) to exist distinctly apart from the CLTAPC and any EMTAPCs that may be present.

Some additional guidance comes by way of the IEEE 1687 IJTAG standard, which provides the following definitions:

"**embedded test access port (eTAP)**: A captive version of a **TAP** that acts as the interface to a portion of the network or a wrapped instrument rather than serving as the device-level TAP. An embedded **TAP** is typically accompanied by an embedded **TAP** controller (**eTAPC**)."

IEEE Std 1687-2014 - Reprinted with permission from IEEE. Copyright IEEE 2014. All rights reserved.





8

"embedded test access port controller (eTAPC): A finite state machine largely corresponding to that specified in IEEE 1149.1 which responds to the embedded TAP signals used as the interface to a sub-network or instrument. To force synchronization, there is a slight restriction from the 1149.1 TAP controller state diagram with respect to the arc traversed when the eTAPC is being re-selected."

IEEE Std 1687-2014 - Reprinted with permission from IEEE. Copyright IEEE 2014. All rights reserved.

The IEEE 1687 IJTAG standard notes further that in some cases an IP provider may deliver an IP block complete with an instrumentation network that is accessed via its own JTAG TAP. When such an IP block is to be integrated into another chip, the block-level TAP becomes an eTAP.

Requirements

Each application domain such as JTAG boundary scan, JTAG software debug and IJTAG instrumentation networks places distinct requirements on a multi-eTAP architecture. The following sections detail the most pertinent of these requirements for each of these three application domains.

JTAG Boundary Scan (IEEE 1149.1-including 1149.4, 1149.6, 1149.8.1-and/or 1149.7)

JTAG boundary scan, as standardized by IEEE 1149.1 and its derivatives, specifies a chip-level instrument infrastructure that is accessed via a chip-level boundary-scan register. This architecture allows test software to affect the states of chip output pins and to detect the states of chip input pins, according to its chief purpose, that of board structural test. This application and, in fact, any based on the conventional use of the TAP are based on certain assumptions about how the TAP presents the chip-level test logic architecture to the higher-level assembly. These assumptions can be retained in multi-eTAP architectures if the following requirements are met.





9

IEEE 1149.1



Figure 1: Conventional test logic architecture defined by IEEE 1149.1

IEEE 1149.1 requires the following behaviors at the chip level:

- The chip shall have one and only one CLTAP (Figure 1), which provides access to chiplevel test logic, including all of the subordinate data registers and eTAPs. The CLTAP consists of the following dedicated pins:
 - Mandatory (4): TCK, TMS, TDI, TDO
 - Optional (1): TRST* (alternatively written as TRST#, TRSTb, TRSTn, etc.).
- The CLTAP operates one and only one TAP controller, a finite state machine conforming to the IEEE 1149.1-specified state diagram (Figure 2). The TAP controller sequences the operation of one instruction register (IR) and multiple test data registers (DR).
- When *Shift-IR* operations are invoked following the *Test-Logic-Reset* state, the chip-level IR will be presented between TDI and TDO and it will have the length that is defined in the chip's Boundary-Scan Description Language (BSDL).
- When the *IDCODE* (or *USERCODE*, if applicable) instruction is invoked following the *Test-Logic-Reset* state, the chip-level DeviceID register will be presented between TDI and TDO for *Shift-DR* operations and it will have a length of 32 bits.



• When the *BYPASS* instruction (*) is invoked following the *Test-Logic-Reset* state, the chip-level Bypass register will be presented between TDI and TDO for *Shift-DR* operations and it will have a length of one bit.

*NOTE: Other IEEE 1149.1-defined instructions that target the Bypass register will be treated similarly. Of course, this includes CLAMP or HIGHZ (if applicable). In addition, when other standards stipulate IEEE 1149.1 compliance, instructions that are defined by those standards and which target the Bypass register will be treated in a similar fashion.

• When a *SAMPLE*, *PRELOAD*, *INTEST* (if applicable) or *EXTEST* instruction (*) is invoked following the *Test-Logic-Reset* state, the chip-level Boundary-Scan Register will be presented between TDI and TDO for *Shift-DR* operations and it will have the length defined in the chip's BSDL.

*NOTE: When other standards stipulate IEEE 1149.1 compliance, the instructions defined by those standards that target the Boundary-Scan Register will be treated in a similar fashion. Of course, this includes PROBE for 1149.4, EXTEST_PULSE and EXTEST_TRAIN for 1149.6, and SELECTIVE_TOGGLE for 1149.8.1.







Figure 2: TAP controller state diagram

With regards to multi-eTAP architectures specifically, the requirements listed above concerning the behavior of *Shift-IR* (or *Shift-DR*) directly following (*) the *Test-Logic-Reset* state can be met by isolating eTAPs from the scan chain when the *Test-Logic-Reset* state is achieved.

*NOTE: On the other hand, it is generally accepted that user-defined instructions such as private instructions may alter registers and their lengths when they are accessed by subsequent Shift-IR or Shift-DR operations. (That is, the Shift-IR or Shift-DR state does not directly follow a Test-Logic-Reset state.)

IEEE 1149.7

In addition to including all of the IEEE 1149.1 functionality explained above, IEEE 1149.7 defines the following behaviors at the chip level:





- Changes to the operating mode of any eTAP shall only take effect upon entering the *Run-Test/Idle* or *Test-Logic-Reset* state (ref IEEE Std 1149.7-2009, sub-clause 16.7.2, rule l):
 - Where the *Run-Test/Idle* state allows for either inclusion or isolation of eTAPs
 - Where the *Test-Logic-Reset* state allows for only isolation of eTAPs.
- When the eTAP selection is controlled by DR, changes to the operating mode of any eTAP will not be affected by any IR operations.
- The preferred method for isolating eTAPs is through TCK gating (ref IEEE Std 1149.7-2009, sub-clause 15.4).

JTAG Software Debug (ARM DAP, Intel ITP and others)

JTAG-based hardware-assisted software debug consists of core-level instrument(s) in the processor that are accessible via various run-control scan registers. This allows debug software to affect processor operating modes and states and to capture processor operating modes and states by way of the Debug Access Port (DAP) defined in the ARM® CoreSight[™] architecture or by way of the In-Target Probe (ITP) for Intel® architecture. This application makes assumptions concerning how the TAP presents the software debug architecture. These assumptions can be retained in multi-eTAP architectures if the following requirements are met.

Conformance to JTAG BYPASS

Tools for software debug applications typically are used interactively in real time by the debug engineer. As such, the tools are implemented so as to operate the run-control scan registers for the most effective and efficient operation of the debug architecture. Long-standing practice dictates that when a given core or core group (where cores are accessed as a group, such as by a DAP) has its TAP or eTAP selected for inclusion in the device scan chain, but is out of scope for a given debug event, the core or core group may be required to operate *BYPASS*, which reduces its effective DR length to one bit.

Compatibility to Automated Discovery

Another long-standing practice in tools for software debug applications is that of automated discovery of the target. Consider the case where a given tool, which comprehends the run-control





facilities offered in a processor core of a given type, may be asked to address targets that are different in the number of cores on a given device. Furthermore, consider the case that a given board-level target might be designed to accept several such device variants. Rather than requiring the debug engineer to manually describe each target configuration, the tool chain can more effectively and efficiently query the target to obtain essential information, such as device count, device types, core counts, core types etc. Long-standing practice dictates that each TAP or eTAP, when initially selected for inclusion in the device scan chain from its *Test-Logic-Reset* state, will operate *IDCODE*, so that it will respond to DR scans with a unique 32-bit identification capture value.

Therefore, for multi-eTAP architectures to maintain conformance to *BYPASS* and compatibility with automated discovery, the following requirements should be met:

- Any eTAP to be operated in the context of a debug (run control) session shall provide an embedded instruction register (eIR) that has a defined fixed length of at least two bits and has a defined capture value, including the mandatory 0b01 value at LSBs (other bits may be defined as don't-care, X).
- Any eTAP to be operated in the context of a debug (run control) session shall provide an embedded Bypass register that has fixed length of one bit, has a defined capture value of 0b0 and is selected by loading any defined opcode for *BYPASS* to the eIR.
- Any eTAP to be operated in the context of a debug (run control) session shall provide a *BYPASS* instruction that has, among the assigned opcodes, at least the all-1s opcode.
- Any eTAP to be operated in the context of a debug (run control) session shall provide a 32bit embedded DeviceID register that has a defined capture value, including the mandatory 0b1 value at LSB, and is selected by loading the defined opcode for *IDCODE* to the eIR.
- Any eTAP to be operated in the context of a debug (run control) session shall provide an *IDCODE* instruction that shall have an arbitrarily defined opcode (as permitted) and shall be loaded to the eIR as an effect of entry to *Test-Logic-Reset* state.

Note that the above requirements may be satisfied in multi-eTAP architectures where the eTAPs are operated in series with each other, but only if they are chained together with no intervening





IEEE 1687 IJTAG Segment Insertion Bits (SIB) or Segment Swap Bits (SSB), because these would cause the apparent length of the eIR to not be fixed.

IJTAG Instrumentation Networks (IEEE 1687 including IEEE 1500)

As specified by IEEE 1687, IJTAG instrumentation networks are comprised of embedded instruments accessed via various network scan registers. This network allows controlling software to alter instrument operating modes and states, and to capture such modes and states for various purposes. This functionality is based on certain assumptions concerning how the TAP presents the instrumentation network architecture. These assumptions can be retained in multi-eTAP architectures if the following requirements are met. Two distinct cases are compared: instruments that are wrapped by a test data register (TDR) versus instruments that are eTAP-connected.

TDR-Wrapped Instruments

Within the context of a design implementing the IEEE 1687 IJTAG standard, an instrument typically may be provided with a simple interface that is suitable for wrapping with a conventional TDR. This TDR can be integral to the IP block that contains the instrument or can be added externally. The register segment for a TDR-wrapped instrument may be attached directly to the CLTAPC and thence provided its own assigned instruction(s) or it may be inserted along with one or more other segments into a hierarchical instrumentation network facilitated by SIBs and/or SSBs.

Even when the one or more SIB/SSB-related instructions associated with a given instrument or segment of an instrumentation network are accessed, resulting in what appears to be a TDR of variable length, the length of the IR remains fixed. Furthermore, the association of an instruction to an instrument or network segment is documented by way of the IJTAG AccessLink, defined in IEEE 1687 as:

"AccessLink: An Instrument Connectivity Language (ICL) keyword that is used to describe the details of the interface between the device pins and the network. The IEEE 1149.1 test access port (TAP) has a well-defined description built into ICL."

IEEE Std 1687-2014 - Reprinted with permission from IEEE. Copyright IEEE 2014. All rights reserved.



SCANWORKS® More visibility. More tools. One Platform. The AccessLink instruction selects a portion of the network to be the active scan chain connected between TDI and TDO. The TAP Controller provides operation signals that sequence in accordance with the IEEE 1149.1 FSM for the selected scan chain. More than one AccessLink instruction may select more than one portion of the IJTAG embedded instrument network and more than one AccessLink instruction may associate different controllers with the network.

Regarding multiple-eTAP architectures, no special requirements pertain to such TDR-wrapped instruments as long as they present as DRs (even those with SIBs and/or SSBs) associated with AccessLink instructions as described here.

eTAP-Connected Instruments

Alternatively, an IJTAG instrument may be provided with a TAP that is integrated into the IP block. This type of instrument falls into two cases.

Partial TAPs (FSM+TDR only)

The case of the partial TAP, which operates without an IR, should be considered degenerate. It does NOT conform to the model of an eTAP as defined by this specification. Its utility derives from its ability to generate scan controls locally, since it has an integral FSM. This reduces the demand on fanout from the CLTAPC. However, as it has no IR and only TDRs, it can be treated as a TDR-wrapped instrument.

Full TAPs (with IR)

On the other hand, the case of an IP block with a full TAP does conform to the model of an eTAP as defined by this specification. Such eTAPs can be retained in multi-eTAP architectures if the following requirements are met:

- Any eTAP selected for inclusion in the device scan chain shall be operated such that its eFSM state remains synchronized to the TAP state of the CLTAPC.
- Furthermore, any eTAP selected for inclusion in the device scan chain shall be concatenated such that it operates its selected DR when the CLTAPC is on the DR-side and operates its IR when the CLTAPC is on the IR-side.



A Modest Proposal for Chip-level Consolidation of a Multiplicity of Embedded TAPs – Revision A

- Any eTAP excluded from the device scan chain shall be parked in the *Test-Logic-Reset* state if the deselection is invoked by reset or in the *Run-Test/Idle* state if the deselection is invoked otherwise.
- Any eTAP excluded from the device scan chain shall, unless reset, hold the IR and DR values present at the time is was deselected.
- In order to facilitate eFSM synchronization, an eTAP may provide a reset input.

Generally speaking, the synchronization of the eFSM in the eTAP with the TAP state in the CLTAPC, especially at time of selection or deselection, is a key implementation concern. As regards time of selection or deselection, the provision for parking in either the *Run-Test/Idle* or *Test-Logic-Reset* state ensures that passing through the *Run-Test/Idle* state after update to the selected mode will allow resumption of synchronized operation with the FSM in the CLTAPC and the eFSMs of other selected eTAPs.

As well, how reset is implemented should require particular attention. A reset input to an eTAP should cause its eFSM state to be immediately matched to that of the CLTAPC even when the eTAP may be deselected. Otherwise, if an asynchronous reset input is not provided, then the eTAP must be synchronized to the CLTAPC by driving TMS to logic-1 for five TCK cycles.

Proposal

The essence of the multi-eTAPs architecture proposed in this specification is that all eTAPs, no matter the application domain, such as JTAG software debug, IJTAG instrumentation networks, embedded core test per IEEE 1500 or some other purpose, are equal peers. No eTAP should be dependent on another. Further, in the case of a failure or defect in one eTAP, the others should continue to be operable.

Figure 3 illustrates the proposed eTAP selection network in one slice. Each eTAPC is accessed by its unique eTAP, designated as eTAP#, where # represents the serial position. The signals on the left-hand side of the figure represent outputs from the CLTAPC, which are required to include or isolate a particular eTAPC in the instrument network.





17



Figure 3: Proposed eTAP selection network (one slice)

Figure 4 is a proposed recommended practice for an architecture of a CLTAPC capable of effectively including or isolating eTAPs in the on-chip scan chain. As such, the signals on the right-hand side of the figure would connect to the like-named signals in Figure 3. Note that TDO1e...TDO(N-1)e and TDI2e...TDINe are not illustrated, as these are chained from one eTAP to the next in this fashion: TDO1e \rightarrow TDI2e...TDO(N-1)e \rightarrow TDINe. On the other hand, the TCKe, TMSe, and RSTe signals are fanned out in broadcast fashion to all eTAPs.

The selection for inclusion or isolation in the scan chain of each eTAP is made by way of the Sel#e signals. These signals originate from the illustrated 'Config' DR. (Throughout the remainder of this specification, this register will be referred to as eTAP_Config.) As illustrated in Figure 4, the Sel#e signals correspond to a one-hot encoding within the eTAP_Config register, such that each eTAP can be individually included or isolated from the scan chain whenever a particular eTAP might be needed or not needed by any application.





A Modest Proposal for Chip-level Consolidation of a Multiplicity of Embedded TAPs – Revision A



Figure 4: Proposed chip-level architecture for multiple eTAPs that are always available (TDO side)

Figure 5 illustrates a CLTAPC architecture that is proposed as an alternative practice for managing the inclusion or isolation of eTAPs from the on-chip scan chain. While sharing most of the attributes of the proposed recommended practice shown in Figure 4, the alternative architecture differs by enabling the global exclusion of eTAPs through an active instruction. These instructions that will exclude the operation of eTAPs will force all Sel#e signals to be inactive by way of the intervening AND gates so as to exclude the eTAPs regardless of the content of the eTAP_Config register.





A Modest Proposal for Chip-level Consolidation of a Multiplicity of Embedded TAPs - Revision A



Figure 5: Alternative chip-level architecture for multiple eTAPs with instruction-limited availability (TDO side)

NOTE: While Figure 3 shows eTAP isolation by TMS-gating, this proposal does not intend to preclude implementations with isolation by TCK-gating. (In fact, for compliance to IEEE 1149.7, TCK-gating is preferred.)

NOTE: While Figure 4 and Figure 5 show eTAP insertion at the TDO side of the chip-level TAP, this proposal does not intend to preclude implementations with eTAP insertion at the TDI side of the chip-level TAP.

NOTE: While Figure 4 and Figure 5 do not show RTI-gating to hold off the effects of changing the eTAPs selection control register at *Update-DR* time, this proposal does not intend to preclude such implementations. (In fact, for compliance to IEEE 1149.7, RTI-gating should be implemented.)

Rules

This proposal for multiple eTAPs architecture considers the context of a SoC that presumes to be compliant to, at least, IEEE 1149.1. In this light, the CLTAPC is defined as the primary, chip-level TAP and TAP controller and all mandatory elements of test logic architecture configured in





the manner required for compliance to IEEE 1149.1. The definition of eTAP has been provided previously in the section titled *Embedded TAP (eTAP) Defined*. With this background, and in light of the proposal overview given above, the following rules pertain.

- Each eTAP shall provide an embedded instruction register (eIR) that has a defined fixed length of at least two bits and has a defined capture value, including the mandatory 0b01 value at LSBs. (Other bits may be defined as don't-care, X.)
- Each eTAP shall provide a one-bit embedded Bypass (eBypass) register that has a defined capture value of 0b0 and is selected by loading any defined opcode for *BYPASS* to the eIR.
- 3) Each eTAP shall provide a 32-bit embedded DeviceID (eDeviceID) register that has a defined capture value, including the mandatory 0b1 value at LSB, and is selected by loading a defined opcode for *IDCODE* to the eIR.
- 4) Each eTAP shall provide a *BYPASS* instruction that shall have at least the all-1s opcode among any other assigned opcode(s).
- 5) Each eTAP shall provide an *IDCODE* instruction that has an arbitrarily defined opcode and shall be loaded to the eIR as an effect of entry to *Test-Logic*-Reset state.
- 6) There must be no variable length or hidden segments in the IR path at any time.
- 7) With particular regard to a device and/or IEEE 1687 IJTAG instrumentation networks:
 - a) Any SIBs and/or SSBs must present only in the DR path and open onto networks or network segments that contain TDR-wrapped instruments associated with a defined AccessLink instruction.
 - b) An IEEE 1687 IJTAG instrument outfitted with the so-called partial TAP, with FSM and TDR(s), but without an IR, shall be treated as a TDR-wrapped instrument.





NOTE: Such partial TAPs do not conform to the definition of an eTAP contained in this proposal and should not be treated as a conforming eTAP.

- 8) The CLTAPC shall provide an eTAP configuration (eTAP_Config) register that has a defined fixed length and is selected by loading a suitable instruction (call it *Config_eTAPs*) provided for that purpose.
- 9) The CLTAPC shall provide a *Config_eTAPs* instruction with an arbitrarily assigned opcode that is documented by means of BSDL.
- 10) The value held in the eTAP_Config register of the CLTAPC shall provide a means for selecting which eTAP(s) are active (selected).

NOTE: The recommended practice is to provide each eTAP (eTAP#) with a distinct Select eTAP signal (Sel#e) and to provide each such signal with its own distinct bit in the eTAP_Config register in what is commonly referred to as one-hot encoding.

- 11) The eTAP_Config register should provide a reset state that causes all Sel#e signals to be inactive. The default reset state by which all eTAPs are deselected can be achieved by:
 - a) asserting TRST*, if present
 - b) entering into the *Test-Logic-Reset* state
 - c) shifting the default value into the eTAP_Config register by operating a DR-scan concluding with Update-DR.
- 12) As previously described, the architecture of Figure 4 presumes that all eTAPs are always available for inclusion in the active scan path strictly on the basis of the value of the eTAP_Config register without any dependence on the instruction operated in the CLTAPC. Within this proposal, this is defined as 'always-available'.
- 13) As previously described, the architecture of Figure 5 presumes that the eTAPs may be available for inclusion in the active scan path as determined by the value of the



eTAP_Config register, but only when a suitable instruction is operated in the CLTAPC. Within this proposal, this is defined as 'instruction-limited'.

- 14) In the case of an instruction-limited architecture, the eTAPs must be available for inclusion in the active scan path at least when the following CLTAPC instructions are operated:
 - a) BYPASS
 - b) IDCODE.

NOTE: All other instructions are optional. The designer must determine whether some instructions will be independent of the eTAP network, but careful consideration should be given as to which instructions limit the availability of eTAPs for inclusion in the scan path.

- 15) When selected, eTAPs operate in series with each other and with the CLTAPC so as to make the multi-eTAP architecture operate as one extended daisy chain with each segment conducting the same FSM-defined operations on the same TCK clock cycles.
- 16) When not selected, unless reset, eTAPs must 'park' in the *Run-Test/Idle* state with their eTDI-eTDO bypassed (multiplexed around as illustrated in Figure 3) so that the registers within the eTAP are not included within the active scan path; further, the eTAP and its registers must 'freeze' (retain their contents) so that they do not process any Shift, Capture, or Update operations.

NOTE: A potential issue of this proposal is that when there is a large number of eTAPs in the deselected state, the combinatorial (unregistered) path through the bypass multiplexers becomes long and may impact scan performance; accordingly, where large numbers of eTAPs must be supported, some mitigation for scan performance may be required.

17) When the CLTAPC enters the *Test-Logic-Reset* state by issuing a global reset, then every eTAP must also process and apply the reset no matter whether any particular eTAP is in





the active scan path or not. To ensure the CLTAPC can successfully reset subordinate eTAPs, the CLTAPC should remain in the *Test-Logic-Reset* state for at least five TCK cycles.

NOTE: As illustrated in Figure 3, the reset or deselected (parked) versus selected (un-parked) condition of an eTAP may be effected by the gating of its eTMS signals. For example, RSTe active forces eTMS high, so the eTAP homes to the *Test-Logic-Reset* state; and Sel#e inactive forces eTMS low, so the eTAP homes to *Run-Test/Idle* state. Such gating may be local to the eTAP or from within the CLTAPC. Generally, to reduce global routing congestion, the local method is preferred.

- 18) When an eTAP is selected, it must be synchronized and brought out of park such that its eFSM state matches that of the CLTAPC, as follows:
 - a) The CLTAPC must exit *Update-IR/DR* using TMS=0 to arrive in the *Run-Test/Idle* state, not the *Select-DR* state.
 - b) If the CLTAPC exits *Update-IR/DR* with TMS=1, the eTAP is NOT selected and it does not come out of park.

NOTE: As a consequence, the eTAP_Config register must have parallel output latches that are held off beyond *Update-IR/DR* until the subsequent entry to the *Run-Test/Idle* state.

- 19) Considering the case where a multi-eTAPs architecture is documented by the means specified in IEEE 1687:
 - a) Describing the association of *Config_eTAPs* instruction to eTAP_Config register by way of an ICL AccessLink instruction may be required.
 - b) Especially with the case of instruction-limited eTAPs, a description of the association of the eTAPs-inclusive instructions to the multi-eTAPs network by way of an ICL AccessLink instruction may be required.





c) Networks that conform to IEEE 1687 IJTAG (that is, those which terminate on TDR-wrapped instruments) require their own ICL AccessLink instructions for each distinct network.

Note that with instruction-limited eTAPs, a network that conforms to IEEE 1687 IJTAG may be operated by eTAPs-inclusive instruction(s). When this is the case, the common instruction(s) would appear both in the AccessLink for the 1687conforming network as well as in the AccessLink for the multi-eTAPs network. Furthermore, for the same 1687-conforming network, some instruction(s) that operate the network may be eTAPs-exclusive, in which case those instruction(s) would not appear in the AccessLink for the multi-eTAPs network.





A Modest Proposal for Chip-level Consolidation of a Multiplicity of Embedded TAPs - Revision A

Conclusions

With the increasing functionality and performance demands placed on them, more and more modern semiconductors are being implemented as SoCs or SiPs. Since most of the IP blocks embedded into SoCs and SiPs feature embedded TAPs (eTAPs), most of the complex devices that are built up from IP blocks end up with multiple eTAPs. These eTAPs are typically implemented for a wide range of purposes, such as JTAG boundary scan, JTAG software debug, IJTAG instrumentation networks and others, all of which have seemingly divergent and potentially conflicting requirements. Nevertheless, it is possible for these eTAPs to coexist behind the same primary chip-level TAP if certain design rules are followed.

The proposal described in this eBook presents an architectural approach that should eliminate conflicts and facilitate coordinated function among multiple eTAPs on the same device.

We Can Help!

ASSET's experts on mixing eTAPs for JTAG, IJTAG, and software debug are ready to share their knowledge and experience with you. We can get you started or guide you through your project. Reach out to us today.



