LUND UNIVERSITY

Accessing Embedded DfT Instruments with IEEE P1687

Larsson, Erik; Ghani Zadegan, Farrokh

Published in:
IEEE 21st Asian Test Symposium (ATS), 2012

DOI:
10.1109/ATS.2012.74

2012

Link to publication

Citation for published version (APA):
Larsson, E., & Ghani Zadegan, F. (2012). Accessing Embedded DfT Instruments with IEEE P1687. In *IEEE 21st Asian Test Symposium (ATS), 2012* (pp. 71-76). IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/ATS.2012.74

Total number of authors:
2

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Download date: 04. May. 2026

Accessing Embedded DfT Instruments with IEEE P1687

Erik Larsson and Farrokh Ghani Zadegan

Abstract—While the advancement in semiconductor technologies enables manufacturing of highly advanced and complex integrated circuits, there is an increasing need of embedded (on-chip) instruments for test, debug, diagnosis, configuration, monitoring, etc. A key challenge is how to access these instruments from chip terminals in a low-cost, non-intrusive, standardized, flexible and scalable manner. The well-adopted IEEE 1149.1 (Joint Test Action Group (JTAG)) standard offers low-cost, non-intrusive and standardized access but lacks flexibility and scalability, which is addressed by the on-going IEEE P1687 (Internal JTAG (IJTAG)) standardization initiative. This paper discusses the need of embedded instrumentation, the shortcomings of IEEE 1149.1 as well as features and challenges of IEEE P1687.

Index Terms—IEEE P1687 IJTAG, IEEE 1149.1, embedded instruments,

1 INTRODUCTION

THE increasing complexity and reduced feature sizes in recent integrated circuit (IC) designs, necessitates on-chip (embedded) instruments for test, debug, diagnosis, configuration, and monitoring. To have an idea of the number of instruments in a modern IC, consider an ASIC from Ericsson which contains 64 cores, each core having its dedicated data and instruction memories [11]. This IC also contains a number of SERDESs and hardware accelerators. Therefore, there are more than 200 blocks of logic inside this ASIC where each block may contain MBIST, LBIST, sensors, etc., which can be regarded as on-chip instruments. It can be seen that the number of instruments in this ASIC would amount up to several hundreds.

There is a need of an access method for these instruments. It is desirable that the access method has *low-cost* in terms of added silicon, is *non-intrusive* such that it does not impact the regular functionality, and is *standardized* to ease integration of instruments from various vendors and to ease access of instruments in the final IC. It is also desirable that the access method is *flexible* such that instruments can be accessed in an arbitrary manner and *scalable* such that a high number of instruments can be added without a too high overhead penalty.

The IEEE Standard 1149.1 (a.k.a. JTAG[1]) [1], originally intended for board test, has proved useful in ad hoc access to such on-chip infrastructure, as discussed in [2]. However, there is a lack of flexibility and scalability. The IEEE P1687 standard proposal [2], [3], [4], [5], [6] aims to address this need of standardization by describing a flexible data transport infrastructure (called *network*) to interface JTAG to the chip internal instruments. P1687 has therefore received the informal name of Internal JTAG (IJTAG). When ratified, P1687 will specify methods for access and control of embedded instruments [3]. Here, instrument refers to any device with a shift-register [7] that could be included in the JTAG scan-path. Examples of instruments include embedded sensors, internal scan-chains and IEEE standard 1500 wrapped cores [8].

P1687 is characterized by introduction of a single JTAG instruction called GateWay ENable (GWEN) and a hardware component called Segment Insertion Bit (SIB). The use of SIBs makes it possible to create a multitude of different networks for the same set of instruments, and to have the benefit of flexibility in scheduling the access to those instruments, as will be discussed in this paper. To setup the scan-path, P1687 proposes to transport SIB control data together with instrument data on a single wire (the JTAG scan-path), and this will affect overall access time (OAT).

Since IEEE P1687 has recently been proposed, only a few studies have considered it [4], [9], [11], [12]. This paper discusses and compares JTAG and IJTAG from a hardware perspective, which is further elaborated on in [11].

2 IEEE 1149.1 (JTAG)

This section describes briefly IEEE 1149.1 (JTAG) [1], also known as Boundary Scan. JTAG was developed to ease the testing of printed circuits boards (PCBs) as those where getting increasingly crowded by ICs. While IEEE 1149.1 is well-adopted, it should be noted that access of embedded instruments is beyond the intended scope.

Fig. 1 shows the conceptual view of the JTAG circuitry in an IC [1]. Access to the on-chip JTAG circuitry is performed through the test access port (TAP). The TAP has four mandatory signals, namely test data input (TDI), test data output (TDO), test mode select (TMS) and test clock (TCK). The TMS signal is decoded by a state diagram (see Fig. 2) to

• E. Larsson is with the Department of Electrical and Information Technology, Lund University, Sweden, e-mail: erik.larsson@eit.lu.se.
• F. Ghani Zadegan is with the Department of Computer and Information Science, Linköping University, Sweden, e-mail: ghanizadegan@ieee.org

1. Joint Test Action Group

1081-7735/12 $26.00 © 2012 IEEE
DOI 10.1109/ATS.2012.74

IEEE computer society

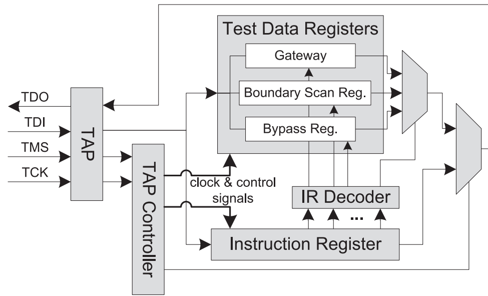


Fig. 1. A conceptual view of JTAG circuitry and how P1687 *Gateway* is interfaced to JTAG

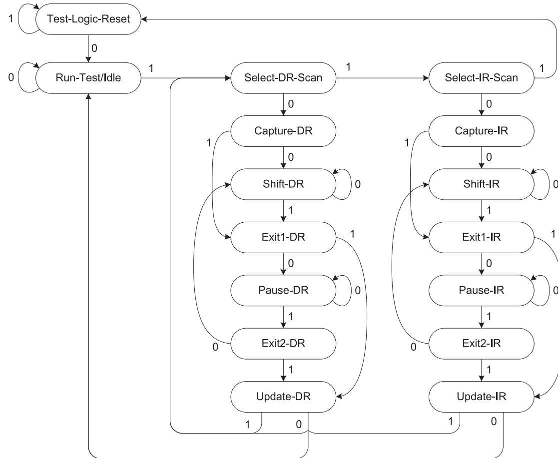


Fig. 2. JTAG TAP Controller state diagram

generate the control signals required for the *capture*, *shift* and *update* operations on instruction register (IR) and test data registers (TDRs). The capture operation is defined as parallel loading of a value into the IR or any of the TDRs, the update operation is defined as transferring logic values from the shift-register stage of the IR or any of the TDRs to their latched parallel outputs, and the *shift* operation is defined as shifting the data serially into and out of the IR or any of the TDRs one bit per TCK [1].

The TAP Controller (Fig. 1) contains a state machine, see Fig. 2. The state diagram has two similar branches, (1) the IR branch used for performing operations on the IR (IR-Scan) and (2) the DR branch used for performing operations on the current TDR (DR-Scan)(Fig. 2). A TDR is selected by the instruction loaded into the IR, which is decoded by the IR Decoder, see Fig. 1.After decoding the instruction, a TDR is selected. The input vectors are shifted serially into the selected TDR by shifting the data when the TAP Controller is in the Shift-DR state. By keeping TMS at logic '0' it is possible to shift in required number of bits. Moving to the Update-DR state makes the shifted vector appear at the parallel outputs of the TDR. The data that should be parallelly loaded into the TDR, i.e. the output vectors, are captured at the Capture-DR state and are shifted out by moving to the Shift-DR state. It is possible to shift in the next input vector while shifting out the output vector corresponding to the previous input vector. The sequence of the five states, Exit1-DR, Update-DR, Select-DR, Capture-DR and Shift-DR, for applying inputs and capturing outputs between two shift operations is denoted a CUC (Cycle of Update and Capture) [11].

JTAG is a successful and widely used standard, and the JTAG TAP is available on most modern ICs [2]. But there are three drawbacks associated with the use of JTAG TAP to access the embedded instruments, namely (1) posing a trade-off between scalability of hardware and flexibility in scheduling the access to the instruments, (2) lack of a language suitable for describing all sorts of instruments, and (3) lack of a language related to the JTAG description that describes the operation of the instrument independent of the placement, configuration, or use of that instrument in the overall access mechanism.

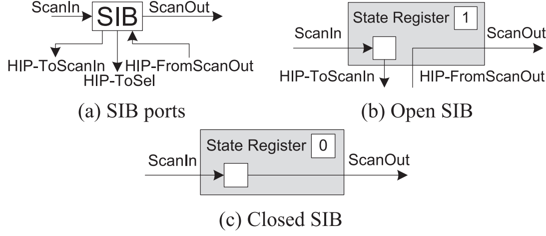
3 IEEE 1149.1 FOR INSTRUMENT ACCESS

While JTAG offers a standardized, low-cost and non-intrusive solution to accessing embedded instruments, there are three drawbacks, which will be detailed in this section.

The first drawback is the lack of flexibility. It can be explained by assuming that each instrument is added to the JTAG circuitry as a separate TDR. In this way, there is a possibility to allow individual access to each of the instruments. However, as only one TDR can be selected at a time, only one instrument can be accessed at a time. Further, the instruction register (IR) becomes very long or the IR Decoder becomes highly complex. If all instruments are chained into a single scan-path (TDR), there is a high penalty in access time when it is desirable to access one instrument only. The scan-chain is fixed, which means that the complete chain of instruments are to be accessed even if access of only one instrument is of interest at the time. In addition, such an architecture has a high risk of failure, since any problem with the scan-path, e.g. stuck-at-fault, will render all instruments in that scan-path inaccessible.

The second drawback with using JTAG to access the instruments is that boundary scan definition language (BSDL) which is part of the JTAG standard and is used to describe the boundary scan devices, is neither efficient nor sufficient to describe all types of instruments [9].

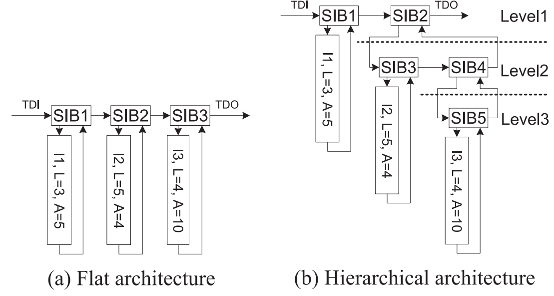
The third drawback with using JTAG is that there is no portable vector or portable procedure language that can be used to describe the operations associated with the instrument regardless of where and how that instrument is used. The serial vector format (SVF) [10], which is used to describe the JTAG operations, is written at the chip-level, not the instrument level. Furthermore, SVF files must be recreated for any changes in the configuration, i.e. length of TAP IR, length of the instrument interface shift-registers and their placement order on the scan-path, etc.



(a) SIB ports (b) Open SIB

(c) Closed SIB

Fig. 3. Simplified view of the SIB component



(a) Flat architecture (b) Hierarchical architecture

Fig. 4. Example P1687 instrument access networks

4 IEEE P1687 (IJTAG)

The IEEE P1687 standard proposal standardizes the way the embedded instruments are accessed through JTAG TAP by (1) proposing an interface between JTAG TAP and on-chip instruments to introduce flexibility and scalability into the JTAG scan-path, (2) proposing an instrument connectivity language (ICL) to describe the characteristics of the instruments and the requirements for interfacing to them [3], and (3) a procedural description language (PDL) to describe the operation of an instrument independent of its placement, to facilitate the re-targeting of the vectors for that instrument to chip-level and board-level. The focus of this paper is on the hardware aspects of P1687.

The interface proposed by P1687 for connecting JTAG TAP to instruments is implemented by adding a TDR called *Gateway* to the JTAG circuitry, see Fig. 1. The Gateway is selected by loading an instruction called GateWay ENable (GWEN) through IR-Scan, which makes the Gateway accessible from the TDI and TDO terminals [1], [3]. Once the GWEN instruction is set, any further access, configuration and control of instruments through P1687 will be done through DR-Scans [1], [3].

P1687 introduces a new hardware component called Segment Insertion Bit (SIB), used to set up the scan-path for P1687 networks. The P1687 Gateway itself is made from one or several SIBs. Fig. 3(a) shows a simplified view of a SIB. Besides the ScanIn and ScanOut terminals, SIB has a hierarchical interface port (HIP) used to connect to a P1687 network segment. A segment can be either simply an instrument or composed of other SIBs. A SIB acts as a doorway with two states. It is either open (Fig. 3(b)) and includes the segment on its HIP (hence the name Segment Insertion Bit), or it is closed (Fig. 3(c)) and transfers the data from its ScanIn port to its ScanOut port, excluding the segment on its HIP.

The hierarchical interface port (HIP) has three terminals: HIP-ToScanIn, HIP-FromScanOut and HIP-ToSel. HIP-ToScanIn and HIP-FromScanOut connect the scan-path to the network segment connected to the HIP when the SIB is open. The HIP-ToSel signal is activated when the SIB is open, to enable the segment connected to the HIP. To clarify the need for HIP-ToSel it should be noted that a SIB, in addition to the terminals shown in Fig. 3(a), has clock, select and three control (i.e. shift-enable, capture-enable and update-enable) input terminals. The clock and control signals are shared among all components in a P1687 network and the control signals should be gated separately for each SIB using its select input. For example, consider that SIB s_1 is to be accessed through the HIP of SIB s_2. In this case, the HIP-ToSel output of s_2 should be connected to the select input of s_1. In a similar way, the assumed interface for the instruments also requires the clock, select, and control inputs in addition to the serial data in and serial data out signals. Here again, the control signals are shared among all instruments (and SIBs) in the network and should be gated for each instrument by using its select input. That is, when an instrument is to be accessed through the HIP of a SIB, the HIP-ToSel output of the SIB should be connected to the select input of that instrument. However, to keep the figures in this paper simple, the HIP-ToSel signal is not shown, but it is assumed that whenever a SIB is open, the instrument or the network segment connected to its HIP is enabled.

The state of SIBs in a P1687 network is set by embedding control bits in each input vector such that after being shifted in, when the TAP Controller is in the Shift-DR state, each control bit is placed into the register of its intended SIB. The control bit for each SIB is then transferred into the SIB's State Register, shown in Fig. 3(b) and Fig. 3(c), once the TAP Controller is in the Update-DR state. Moving to Update-DR and back to Shift-DR for shifting out the output vector and shifting in the next input vector is part of the cycle of update and capture (CUC) explained in Section 3.

Fig. 4(a) shows a P1687 network of three instruments (I1, I2 and I3) and three SIBs, one for each instrument. The control, select and clock signals are not shown. In Fig. 4(a), L stands for the length of the shift-register for each instrument and A stands for the number of instrument-specific *accesses*. In this paper, *access* is defined as (1) shifting input bits into the instrument's shift-register, (2) latching the contents of the shift-register to be applied to the internal circuitry of the instrument, (3) capturing the output of the instrument into the shift-register and (4) shifting the captured values out. The shifting-out of the instrument outputs can overlap in time with shifting-in the input command bits for the next access. Considering the relatively slow P1687 clock (i.e. TCK applied to JTAG TAP) [4], [5], it is assumed that the time it takes

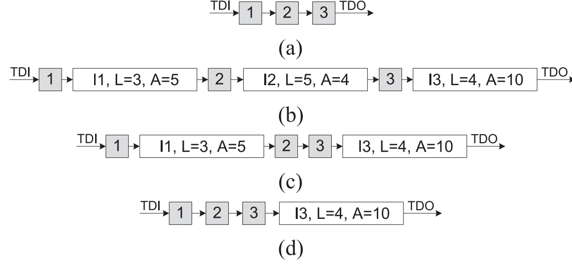


Fig. 5. Scan-path configurations of the flat architecture example for the concurrent schedule

an instrument to process the applied inputs and make the outputs ready to be captured, is less than the time it takes to move from Update-DR to Capture-DR in the TAP Controller.

It is important to note that not all instrument types are accessed as described above. For example, a BIST engine might be selected (by opening its corresponding SIB) and activated (by launching the BIST) and then be de-selected (by closing its SIB) while still active and running. Later in the access schedule, the BIST can be selected again and its Done and Fail signals be polled.

The type of architecture that is implemented by the SIBs in Fig. 4(a), is called a flat architecture in the remainder of this paper. In the flat architecture no SIB is connected to the HIP of another SIB. Fig. 4(b) shows another network of the same three instruments, i.e. I1, I2, and I3. Here, there are five SIBs and three of these SIBs are connected to the TAP through the HIP of SIB2. This type of architecture is called hierarchical architecture in the remainder of this paper. Each SIB that has another SIB connected to its HIP, represents a doorway to another level of hierarchy, such as SIB2 and SIB4 in Fig. 4(b). In this paper, for the sake of terminology, a SIB having only an instrument on its HIP is referred to as an *instrument SIB* and a SIB having one or more SIBs on its HIP is called a *doorway SIB*. It should be noted that SIB1, SIB2 and SIB3 in Fig. 4(a) form the IJTAG Gateway, while in Fig. 4(b) SIB1 and SIB2 form the Gateway. The SIBs forming the Gateway receive their select signal directly from JTAG instruction decoder (IR Decoder in Fig. 1).

5 IEEE P1687 FOR INSTRUMENT ACCESS

In this section, the access to instruments is studied for the flat architecture and the hierarchical architecture. For each architecture, concurrent and sequential access to the instruments are studied. The architectures with instruments (register lengths (L) and number of accesses (A)) are detailed in Fig. 4.

5.1 Flat architecture

5.1.1 Concurrent schedule

In the concurrent schedule, accesses for all instruments start as soon as possible, which for the flat architecture means all accesses start at the same time.

TABLE 1
Flat architecture, concurrent schedule

Row No.	Scan-path	Scanned bits: SIBs	I1	I2	I3	CUC	# of scan sequences	Sum for scan-path
1	Fig. 5(a)	3	0	0	0	5	1	$3+5$
2	Fig. 5(b)	3	3	5	4	5	5	$(15+5)\cdot 5$
3	Fig. 5(c)	3	3	0	4	5	1	$10+5$
4	Fig. 5(d)	3	0	0	4	5	5	$(7+5)\cdot 5$
OAT								$\sum$=183

When an instrument is no more active (i.e. there are no more inputs to be applied to it) it is excluded from the network, by closing its corresponding instrument SIB. This makes the scan-path shorter for accessing the rest of the instruments. Hence, the overall access time is reduced. This type of concurrency is not possible using original JTAG specifications and is unique to P1687 in this regard.

The input data for the instruments and the control bits for the SIBs on the current scan-path are concatenated appropriately to form the input vector. While an input vector is shifted in, the output vector corresponding to the previous inputs to the instruments is shifted out. Each output vector contains the output data from each of the instruments on the scan-path and the contents of the registers of the SIBs on the scan-path.

In the following, it will be described how to calculate the access time for the flat architecture shown in Fig. 4(a) and the concurrent schedule, with the help of Fig. 5 and Table 1. In Fig. 5, the gray boxes represent the registers inside the correspondingly numbered SIBs.

Before accessing the instruments, the SIBs must be opened, since the scan-path initially only consists of the SIBs in the Gateway, as shown in Fig. 5(a). To open the SIBs, three bits with logic value of '1' are scanned in (one bit for each SIB) and subsequently a CUC is performed. The three bits each corresponds to the register of a closed SIB, and they are accounted for on the row marked 1 in Table 1, column "SIBs". After the CUC, which takes five clock cycles (TCKs) as indicated in the column "CUC", all instruments are included in the scan-path, as shown in Fig. 5(b). At this point, input data can be applied to all three instruments, with a total scan-path length of 1_{SIB1}+3_{I1}+1_{SIB2}+5_{I2}+1_{SIB3}+4_{I3}=15 bits, where 1_{SIBx} represents the 1-bit register inside SIB$_x$. The number of bits for the three instruments (called 3_{I1}, 5_{I2}, 4_{I3} above) are counted in the columns I1, I2 and I3 of Table 1. After four input vectors have been applied, accessing instrument I2 is complete and its shift-register should be excluded from the scan-path, which is done by setting the control bits such that SIB2 is closed, and SIB1 and SIB3 are kept open. This operation, to close SIB2, cannot occur until the output corresponding to the last input to I2 has been scanned out. Therefore, a fifth scan sequence is required during which the last output vector of I2 is scanned out and the SIB control bits to exclude I2 from the scan-path are scanned in. In total, five scan sequences are performed on the scan-path shown in

TABLE 2
Flat architecture, sequential schedule

Row No.	Scan-path	Scanned bits SIBs	I1	I2	I3	CUC	# of scan sequences	Sum for scan-path
1	Fig. 6(a)	3	0	0	0	5	1	$3+5$
2	Fig. 6(b)	3	3	0	0	5	6	$(6+5)\cdot 6$
3	Fig. 6(c)	3	0	5	0	5	5	$(8+5)\cdot 5$
4	Fig. 6(d)	3	0	0	4	5	11	$(7+5)\cdot 11$
OAT								$\sum$=271

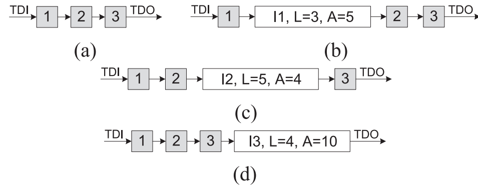


Fig. 6. Scan-path configurations of the flat architecture example for the sequential schedule

Fig. 5(b), which is represented under column "# of scan sequences" in the row marked 2. After exclusion of I2 from the network, the scan-path has a total length of 1_{SIB1}+3_{I1}+1_{SIB2}+1_{SIB3}+4_{I3}=10 bits. The scan-path is now as shown in Fig. 5(c). After one scan sequence which is shifting out the last outputs of I1, represented by the row marked 3, the access to instrument I1 is complete and SIB1 is closed. The scan-path becomes as shown in Fig. 5(d). Four input vectors, hence four scan sequences, remain for instrument I3 and one more scan sequence is used to scan out the last of the outputs for instrument I3, while closing SIB3. For these last five scan sequences the total scan-path length is 1_{SIB1}+1_{SIB2}+1_{SIB3}+4_{I3}=7 bits, as shown in the row marked 4.

Table 1 shows the number of bits of different types that are scanned in for each scan sequence and the number of sequences performed on each scan-path configuration. The scan-path configuration corresponding to each row is specified under the column "Scan-path". The last column (i.e. "Sum for scan-path") shows the total number of bits that are scanned in for each scan-path. OAT is the sum of the values in this last column, as shown on the last row, which for this example is 183 clock cycles.

Note that the SIB control bits contribute to access time by $3+3\cdot 5+3+3\cdot 5 = 36$ clock cycles. Furthermore, the number of clock cycles spent performing CUC is $5+5\cdot 5+5+5\cdot 5 = 60$. These 36+60=96 clock cycles spent scanning SIB control bits and performing CUC are considered overhead, because no actual instrument data is transported during this time.

It can be seen that OAT consists of three components, namely instrument data, SIB programming overhead, and CUC overhead. An *overhead ratio* can be defined as total overhead divided by the access time, which for the above example is calculated as $96/183 \approx 0.52$.

5.1.2 Sequential schedule

In this section, access time will be calculated for the flat architecture considering the sequential schedule. In the sequential schedule, the instruments are accessed one at a time, and the assumed order of access is the order that the instruments appear on the scan-path when all SIBs are open. It is also assumed that the access for each instrument is completed before accessing any other instrument.

Similar to how Fig. 5 and Table 1 described the access for the concurrent schedule, Fig. 6 and Table 2 will be used to explain the steps of sequential access to the instruments in the network shown in Fig. 4(a). Initially, the scan-path is as shown in Fig. 6(a). Three bits are used in the first scan sequence (see the row marked 1 in Table 2) to open SIB1 so that for the six following scan sequences the scan-path is as shown in Fig. 6(b). The row marked 2 in Table 2 shows that the three bits of I1 are included in scan-path. After six scan sequences (see the row marked 2), all the five input vectors for I1 have been applied and the corresponding outputs have been scanned out, while closing SIB1 and opening SIB2 so that the scan-path becomes as shown in Fig. 6(c). For this configuration of the scan-path, four input vectors for I2 are applied followed by a scan sequence to scan out the last outputs (see the row marked 3). Fig. 6(d) shows the scan-path as it is after completing the access for I2. Finally, 11 scan sequences (see the row marked 4) are applied to complete the access for I3 and scan out the last outputs, while closing SIB3. As can be seen from Table 2, access time for the sequential schedule is 271 clock cycles, which should be compared to 183 clock cycles for the concurrent schedule discussed in Table 1. The difference in OAT can be explained by a larger number of scan sequences performed in the sequential schedule, which leads to more SIB and CUC overheads. Similar to how the overhead ratio was calculated in Section 5.1.1, the overhead ratio for the flat architecture and the sequential schedule can be calculated as $(69+115)/271 \approx 0.68$.

5.2 Hierarchical architecture

This section discusses the overall access time (OAT) for the hierarchical architecture shown in Fig. 4(b).

Table 3 and Table 4 show the steps to calculate OAT for the concurrent and sequential schedules, respectively. These tables are similar to Tables 1 and 2 in structure. The possible configurations referred to by the column "Scan-path", are presented in Fig. 7.

The access according to the concurrent schedule was explained for the flat architecture in Section 5.1.1. For the hierarchical architecture, in contrast to the flat architecture, when all instruments in a network segment have become inactive, the doorway SIB whose HIP is connected to that segment will be closed to exclude all the instruments and SIBs on that segment from the scan-path. As can be seen from Table 3, OAT for the hierarchical architecture and the concurrent schedule is 223 clock cycles, which should be compared to 183 clock cycles for the concurrent schedule and the flat architecture. In this example, the hierarchical architecture leads to a longer OAT because of

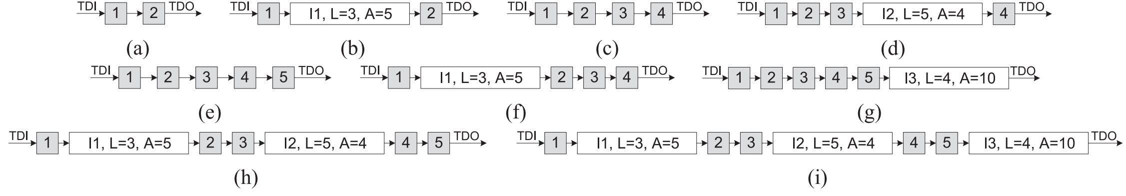


Fig. 7. Scan-path configurations of the hierarchical architecture example

TABLE 3
Hierarchical architecture, concurrent schedule

Row No.	Scan-path	Scanned bits SIBs	I1	I2	I3	CUC	# of scan sequences	Sum for scan-path
1	Fig. 7(a)	2	0	0	0	5	1	$2+5$
2	Fig. 7(f)	4	3	0	0	5	1	$7+5$
3	Fig. 7(h)	5	3	5	0	5	1	$13+5$
4	Fig. 7(i)	5	3	5	4	5	4	$(17+5)\cdot 4$
5	Fig. 7(g)	5	0	0	4	5	7	$(9+5)\cdot 7$
OAT								$\sum$=223

TABLE 4
Hierarchical architecture, sequential schedule

Row No.	Scan-path	Scanned bits SIBs	I1	I2	I3	CUC	# of scan sequences	Sum for scan-path
1	Fig. 7(a)	2	0	0	0	5	1	$2+5$
2	Fig. 7(b)	2	3	0	0	5	6	$(5+5)\cdot 6$
3	Fig. 7(c)	4	0	0	0	5	1	$4+5$
4	Fig. 7(d)	4	0	5	0	5	5	$(9+5)\cdot 5$
5	Fig. 7(e)	5	0	0	0	5	1	$5+5$
6	Fig. 7(g)	5	0	0	4	5	11	$(9+5)\cdot 11$
OAT								$\sum$=310

two factors. Firstly, the overhead from the additional SIBs affects access time. Secondly, the overhead in terms of capture-and-update cycles (CUC) is higher, due to opening the doorway SIBs to access the other levels of hierarchy.

Accessing instruments according to the sequential schedule was discussed in Section 5.1.2. For the hierarchical architectures, it is additionally assumed that only those doorway SIBs are open which are on the shortest scan-path to the instrument being accessed. Table 4 shows that for the sequential schedule, access time is 310 clock cycles, which should be compared with 271 clock cycles for the sequential schedule and the flat architecture. The reason for the higher access time with the hierarchical architecture is more SIB programming overhead and more CUCs.

The overhead ratio for the hierarchical architecture can be calculated as $(66 + 70)/223 \approx 0.61$ for the concurrent schedule and $(98 + 125)/310 \approx 0.72$ for the sequential schedule.

It should be noted that in the example discussed in Section 5 and this section, the flat architecture and the concurrent schedule led to the lowest access time. However, this is not a general conclusion, since other examples may show lower access time on other architectures and schedules. For example, if for the networks shown in Fig. 4 the number of accesses for instruments were 20, 5 and 2 for I1, I2 and I3 respectively, the hierarchical network resulted in a lower access time for both concurrent and sequential schedules.

6 Conclusion

There is an increasing number of embedded (on-chip) instrumentation for test, debug, diagnosis, configuration, monitoring, etc., which enforces an access methodology that is low-cost, non-intrusive, standardized, flexible and scalable. As, the existing IEEE 1149.1 meets all but flexibility and scalability, IEEE P1687 IJTAG is proposed. This paper compares IEEE 1149.1 and IEEE P1687.

References

[1] IEEE association, "IEEE Std 1149.1-2001, IEEE Standard Test Access Port and Boundary-Scan Architecture," 2001.

[2] J. Rearick, B. Eklow, K. Posse, A. Crouch, and B. Bennetts, "IJTAG (Internal JTAG): A Step Toward a DFT Standard," in *Proc. ITC*, 2005.

[3] IJTAG, "IJTAG - IEEE P1687," 2010, http://grouper.ieee.org/groups/1687.

[4] J. Rearick and A. Volz, "A Case Study of Using IEEE P1687 (IJTAG) for High-Speed Serial I/O Characterization and Testing," in *Proc. ITC*, 2006, pp. 1–8.

[5] K. Posse, A. Crouch, J. Rearick, B. Eklow, M. Laisne, B. Bennetts, J. Doege, M. Ricchetti, and J.-F. Cote, "IEEE P1687: Toward Standardized Access of Embedded Instrumentation," in *Proc. ITC*, 2006, pp. 1 –8.

[6] A. L. Crouch, "IJTAG: The Path to Organized Instrument Connectivity," in *Proc. ITC*, 2007, pp. 1–10.

[7] J. Doege and A. Crouch, "The advantages of limiting p1687 to a restricted subset," in *Proc. ITC*, 2008, pp. 1 –8.

[8] IEEE association, "IEEE Std 1500-2005, IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits," 2005.

[9] M. Portolan, S. Goyal, B. Van Treuren, C.-H. Chiang, T. Chakraborty, and T. Cook, "A common language framework for next-generation embedded testing," *Design & Test of Computers, IEEE*, vol. 27, no. 5, pp. 36 –49, 2010.

[10] ASSET InterTech, Inc., "Serial Vector Format Specification," 1999, http://www.asset-intertech.com/support/svf.pdf.

[11] F. G. Zadegan, U. Ingelsson, G. Carlsson, and E. Larsson, "Access Time Analysis for IEEE P1687," in *Computers, IEEE Transactions on*, vol. 61, nr. 10, pp. 1459–1472, october 2012.

[12] F. Ghani Zadegan, U. Ingelsson, E. Larsson, G. Carlsson, "Reusing and Retargeting On-Chip Instrument Access Procedures in IEEE P1687," in *Design Test of Computers, IEEE*, vol. 29, nr. 2, pp. 79–88, april 2012.