

## 194 Changes introduced by this revision

195 First, this version of the standard affirms what had been required in the previous version. There are only minor  
196 clarifications or relaxations to the rules that are already established.

197 Second, while this is a major revision, items introduced in this version are optional and intended to provide test  
198 improvements in the complex components being created today and in the foreseeable future. There are also  
199 significant improvements in documentation capability, including the introduction of the ability to document test  
200 procedures unique to the component.

201 The major new capabilities are listed in the order in which they appear.

202 In the standard body:

- 203 • A new, optional, Test Mode Persistence controller that can maintain the 1149.1 test logic in test mode and  
204 the system logic in its “safe and cool” state. Clause 6 is now split into Clause 6.1 for the TAP controller  
205 and Clause 6.2 for the Test Mode persistence controller. In support of this new controller, there are two  
206 new instructions CLAMP\_HOLD in Clause 8.20, with its new bypass-escape test data register in Clause  
207 16, and CLAMP\_RELEASE in Clause 8.20.
- 208 • A new, optional ECIDCODE instruction in Clause 8.15 and its ecid test data register in Clause 13 to  
209 supplement the existing IDCODE and USERCODE instructions and allow the recovery of an Electronic  
210 Chip ID value used to identify and track individual chips.
- 211 • A new, optional, chip initialization capability to provide more flexibility in preparing the component for  
212 test. The INIT\_SETUP and INIT\_RUN instructions in Clauses 8.17, 8.18, and 8.19, and their new  
213 init\_data and init\_status test data registers in Clauses 14 and 15, respectively. This will allow  
214 programmable I/O to be set up prior to board or system testing, as well as any tasks required to put the  
215 system logic into a “safe and cool” state for test.
- 216 • A new, optional, IC\_RESET instruction in Clause 8.21 and its reset\_select test data register in Clause 17 to  
217 provide control of component reset functions through the TAP.
- 218 • In Clause 9.2, an optional TAP to test data register interface is recommended, and examples of different  
219 types of test data register cells using this interface are shown. In addition, the concept of register segments  
220 is expanded to allow segments that may be excluded or included. This is introduced to support power  
221 domains that may be powered down, and yet may have a segment of a test data register within that domain.  
222 However, the capability was kept general.
- 223 • In the new Clause 9.4, the rules for defining and controlling the new excludable segments are established.
- 224 • Boundary-scan register definition in Clause 11 has been updated to support:
  - 225 ○ optional excludable boundary-scan register segments;
  - 226 ○ optional placement of observe-only boundary-scan register cells on all pins except the TAP pins;
  - 227 ○ optional placement of fault observe-only boundary-scan register cells on all pins except the TAP  
228 pins.
- 229 • Documentation requirements in Clause 18 have been updated for the new capabilities.

230 In the Boundary Scan Description Language (BSDL) in Annex B:

- 231 • The entire Annex was rewritten for:
  - 232 ○ Increased clarity of what was normative vs. descriptive text, and
  - 233 ○ Increased consistency in presentation;
- 234 • BSDL is no longer a “proper subset” of VHDL, but rather is now “based on” VHDL. See Clause B.4. In  
235 particular, new port type keywords were introduced in Clause B.8.3 that are not needed in VHDL, but give  
236 a more accurate description of each port in BSDL;
- 237 • Formal definitions of language elements are included in Clause B.5 instead of reliance on inheritance from  
238 VHDL;
- 239 • Some changes to the BNF notation used, including definition of all the special character tokens, is in  
240 Clause B.6;

- 241
- 242
- 243
- 244
- 245
- 246
- 247
- 248
- 249
- 250
- 251
- Pin mapping in Clause B.8.7 now allows documenting that a port is not connected in a specific package;
  - The boundary-scan register description in Clause B.8.14 introduces new attributes for defining boundary-scan register segments, and introduces a requirement for documenting the behavior of an undriven input;
  - New capabilities are introduced for documenting the structural details of test data registers:
    - Clause B.8.17 introduces the definition of mnemonics that may be associated with register fields;
    - Clause B.8.18 introduces the ability to name fields within a register or segment, and the ability to hierarchically assemble segments into larger segments or whole registers;
    - Clause B.8.19 introduces the ability to associate a register field or bit to specific ports, and to associate a power port to other ports;
  - The role of a User Defined Package file defined in Clause B.10 has been expanded to support logic IP providers who may need to document test data register segments contained within their IP.

252 A new Annex C codifies the Procedural Description Language (PDL), a new language for documenting the  
253 procedural and data requirements for some of the new instructions. As mentioned above, this version of the  
254 standard introduces new instructions for configuring complex I/Os prior to entering the EXTEST instruction. As  
255 the data required by the *INIT\_SETUP* instruction could vary for each instance of the component on each distinct  
256 board or system design, this created the need for a new language for setting internal TDR register fields in order to  
257 configure the I/O.

258