IEC 62014-4

Edition 2.0 2025-06

INTERNATIONAL STANDARD

IEEE Std 1685™

IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows

ICS 35.240.50; 25.040.01

ISBN 978-2-8327-0468-4

IEC 62014-4:2025-06(en) IEEE Std 1685-2022

THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2022 IEEE

All rights reserved. IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the IEC Central Office. Any questions about IEEE copyright should be addressed to the IEEE. Enquiries about obtaining additional rights to this publication and other information requests should be addressed to the IEC or your local IEC member National Committee.

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue
New York, NY 10016-5997
United States of America
stds.ipr@ieee.org
www.ieee.org

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC/IEEE publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews, graphical symbols and the glossary. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 500 terminological entries in English and French, with equivalent terms in 25 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

Warning! Make sure that you obtained this publication from an authorized distributor.

Contents

1. Overview ... 19

1.1 Scope ... 19
1.2 Purpose ... 19
1.3 Word usage ... 20
1.4 Design environment ... 20
1.5 IP-XACT–enabled implementations ... 25
1.6 Conventions used ... 26
1.7 Use of color in this standard ... 31
1.8 Contents of this standard ... 31

2. Normative references ... 33

3. Definitions, acronyms, and abbreviations ... 34

3.1 Definitions ... 34
3.2 Acronyms and abbreviations ... 39

4. Interoperability use model ... 40

4.1 Roles and responsibilities ... 40
4.2 IP-XACT IP exchange flows ... 41

5. Interface definition descriptions ... 43

5.1 Definition descriptions ... 43
5.2 Bus definition ... 43
5.3 Abstraction definition ... 45
5.4 Ports ... 46
5.5 Wire ports ... 47
5.6 Qualifiers ... 49
5.7 Wire port group ... 51
5.8 Wire port mode (and mirrored mode) constraints ... 52
5.9 Transactional ports ... 53
5.10 Transactional port group ... 54
5.11 Packets ... 55
5.12 Extending bus and abstraction definitions ... 56
5.13 Clock and reset handling ... 60

6. Component descriptions ... 61

6.1 Component ... 61
6.2 Type definitions ... 63
6.3 Power domains ... 65
6.4 Interfaces ... 66
6.5 Interface interconnections ... 67
6.6 Complex interface interconnections ... 68
6.7 Bus interfaces ... 71
6.8 Indirect interfaces ... 79
6.9 Component channels ... 81
6.10 Modes ... 82

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

6.11 Address spaces 83
6.12 Memory maps 87
6.13 Remapping 99
6.14 Registers 100
6.15 Models 115
6.16 Component generators 151
6.17 File sets 153
6.18 Clear box elements 159
6.19 Clear box element reference 161
6.20 CPUs 162
6.21 Reset types 167

7. Design descriptions 168

7.1 Design 168
7.2 Design component instances 170
7.3 Design interconnections 171
7.4 Active, hierarchical, monitored, and monitor interfaces 173
7.5 Design ad hoc connections 176
7.6 Port references 178

8. Abstractor descriptions 180

8.1 Abstractor 180
8.2 Abstractor interfaces 182
8.3 Abstractor models 183
8.4 Abstractor views 184
8.5 Abstractor ports 185
8.6 Abstractor wire ports 186
8.7 Abstractor transactional port 189
8.8 Abstractor structured ports 190
8.9 Abstractor generators 193

9. Type definitions descriptions 194

9.1 Type definitions 194
9.2 Field access policy definition 197
9.3 Enumeration definition 198
9.4 Field definition 199
9.5 Register definition 200
9.6 Register file definition 201
9.7 Address block definition 202
9.8 Bank definition 203
9.9 Memory map definition 205
9.10 Memory remap definition 207

10. Generator chain descriptions 208

10.1 generatorChain 208
10.2 generatorChainSelector 210
10.3 generatorChain component selector 211
10.4 generatorChain generator 212

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

11. Design configuration descriptions 214

11.1 Design configuration 214
11.2 designConfiguration 215
11.3 interconnectionConfiguration 216
11.4 abstractorInstance 217
11.5 viewConfiguration 218

12. Catalog descriptions 219

12.1 catalog 219
12.2 ipxactFile 221

13. Addressing 222

13.1 Calculating the bit address of a bit in a memory map or address space 222
13.2 Calculating the bus address at the bus interface 224
13.3 Calculating the address at the indirect interface 224
13.4 Address modifications of a channel 225
13.5 Address translation in a bridge 225

14. Data visibility 226

14.1 Mapped address bits mask 226
14.2 Address modifications of an interconnection 226
14.3 Bit steering in a channel 226
14.4 Visibility of bits 227

Annex A Bibliography 230

Annex B Semantic consistency rules 231

B.1 Semantic consistency rule definitions 231
B.1.1 Compatibility of busDefinitions 231
B.1.2 Interface mode of a bus interface 231
B.1.3 Compatibility of abstractionDefinitions 231
B.1.4 Element referenced by configurableElementValue element 231
B.1.5 Memory mapping 231
B.1.6 Port connection equivalence class 232
B.1.7 Logical and physical ports 232
B.1.8 Addressable bus interface 232
B.1.9 Field connection graph 232
B.2 Rule listings 234
B.2.1 Cross-references and VLNVs 234
B.2.2 Interconnections 238
B.2.3 Channels, bridges, and abstractors 240
B.2.4 Monitor interfaces and monitor interconnections 243
B.2.5 Configurable elements 244
B.2.6 Ports 247
B.2.7 Registers 255
B.2.8 Memory maps 259
B.2.9 Addressing 260
B.2.10 Hierarchy 261
B.2.11 Hierarchy and memory maps 262

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

B.2.12 Constraints 262
B.2.13 Design configurations 264
B.2.14 Expressions 265
B.2.15 Access handles 268

Annex C Common elements and concepts 269

C.1 accessHandles 269
C.1.1 simpleAccessHandle 269
C.1.2 slicedAccessHandle 270
C.1.3 portAccessHandle 271
C.1.4 sliceType 272
C.1.5 portSliceType 272
C.2 accessPolicies 273
C.2.1 Schema 273
C.2.2 Description 273
C.3 Access resolution 274
C.4 arrays 275
C.4.1 Configurable arrays with arrayId 275
C.4.2 Configurable arrays without arrayId 276
C.4.3 Memory arrays with stride 276
C.4.4 Field arrays with bit stride 277
C.5 assertions 278
C.5.1 Schema 278
C.5.2 Description 278
C.6 attributes 279
C.6.1 Schema 279
C.6.2 Description 280
C.7 complexBaseExpression 281
C.7.1 complexTiedValueExpression 282
C.7.2 qualifiedExpression 282
C.7.3 realExpression 283
C.7.4 signedLongintExpression 284
C.7.5 stringExpression 285
C.7.6 unresolvedStringExpression 285
C.7.7 unresolvedUnsignedBitExpression 286
C.7.8 unresolvedUnsignedPositiveIntExpression 286
C.7.9 unsignedBitExpression 287
C.7.10 unsignedBitVectorExpression 287
C.7.11 unsignedIntExpression 288
C.7.12 unsignedLongintExpression 289
C.7.13 unsignedPositiveIntExpression 290
C.7.14 unsignedPositiveLongintExpression 291
C.8 choices 292
C.8.1 Schema 292
C.9 configurableElementValues 293
C.9.1 Schema 293
C.9.2 Description 293
C.10 configurableLibraryRefType 294
C.10.1 Schema 294
C.10.2 Description 294
C.11 documentName group 295
C.11.1 Schema 295
C.11.2 Description 295

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

C.12 Endianness 296
C.13 fieldReferenceGroup 297
C.13.1 Schema 297
C.13.2 Description 297
C.14 fieldSliceReferenceGroup 299
C.14.1 Schema 299
C.14.2 Description 300
C.15 fileSetRef 300
C.15.1 Schema 300
C.15.2 Description 301
C.16 fileType 301
C.16.1 Schema 301
C.16.2 Description 301
C.17 indices 302
C.17.1 Schema 302
C.17.2 Description 302
C.18 libraryRefType 303
C.18.1 Schema 303
C.18.2 Description 303
C.19 Name groups 304
C.19.1 nameGroup group 304
C.19.2 nameGroupNMTOKEN group 305
C.19.3 nameGroupOptional group 306
C.19.4 nameGroupPort group 307
C.19.5 nameGroupString group 308
C.20 nameValuePairType 308
C.20.1 Schema 308
C.20.2 Description 309
C.21 parameters 309
C.21.1 Schema 309
C.21.2 Description 309
C.22 partSelect 310
C.22.1 Schema 310
C.22.2 Description 310
C.23 pathSegments 311
C.23.1 pathSegment 311
C.23.2 portPathSegment 311
C.24 Power constraints 312
C.24.1 transactionalPowerConstraints 312
C.24.2 wirePowerConstraints 312
C.25 range 313
C.25.1 Schema 313
C.25.2 Description 313
C.26 Vectors 314
C.26.1 vectors 314
C.26.2 extendedVectorsType 314
C.27 vendorExtensions 315
C.27.1 Schema 315
C.27.2 Description 315
C.28 versionedIdentifier group 315
C.28.1 Schema 315
C.28.2 Description 316
C.28.3 Sorting and comparing version elements 316
C.28.4 Version control 317

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

C.29 viewRef ... 318
C.29.1 Schema ... 318
C.29.2 Description ... 318
C.30 xml:id ... 318
C.31 Component vs. Abstraction Definition Ports ... 318

Annex D Types ... 320

D.1 boolean ... 320
D.2 float ... 320
D.3 ID or IDREF ... 320
D.4 instancePath ... 320
D.5 integer ... 320
D.6 libraryRefType ... 320
D.7 Name ... 320
D.8 NMTOKEN ... 321
D.9 NMTOKENS ... 321
D.10 portName ... 321
D.11 ipxactURI ... 321
D.12 string ... 321
D.13 token ... 321

Annex E SystemVerilog expressions ... 322

E.1 Overview ... 322
E.2 Data-types ... 322
E.2.1 bit data type ... 322
E.2.2 byte data type ... 322
E.2.3 shortint data type ... 323
E.2.4 int data type ... 323
E.2.5 longint data type ... 323
E.2.6 shortreal data type ... 323
E.2.7 real data type ... 323
E.2.8 string data type ... 323
E.2.9 Signed and unsigned data types ... 324
E.2.10 Unresolved data types ... 324
E.3 Assignments ... 324
E.3.1 Single value assignment ... 324
E.3.2 Parameter type ... 324
E.3.3 Parameter signing ... 325
E.3.4 Vector assignment ... 325
E.3.5 Array assignment ... 325
E.3.6 Identifiers ... 326
E.3.7 Identifier references ... 326
E.4 Operators ... 328
E.5 Functions ... 329
E.5.1 Integer function ... 329
E.5.2 Real functions ... 329
E.5.3 String function ... 330
E.5.4 IP-XACT specific functions ... 332
E.5.5 IP-XACT specific escape sequences ... 334
E.6 Expression language formal syntax (BNF) ... 335
E.6.1 Declarations: declaration data types ... 335
E.6.2 Behavioral statements: Case statements ... 335

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

E.6.3 Expressions ... 336
E.6.4 General: Identifiers ... 337
E.7 SystemVerilog conversion steps ... 338
E.7.1 Convert parameter ... 338
E.7.2 Convert expression ... 338
E.8 SystemVerilog reference ... 338

Annex F Tight generator interface ... 341

F.1 Method of communication ... 341
F.2 Generator invocation ... 341
F.2.1 Resolving the URL ... 342
F.2.2 Example ... 342
F.3 TGI API ... 343
F.3.1 TGI fault codes ... 344
F.3.2 Administrative commands ... 344
F.3.3 Return values ... 345
F.4 IDs and configurable values ... 345
F.5 TGI messages ... 346
F.6 Vendor attributes ... 346
F.7 TGI calls ... 346
F.7.1 Category index ... 346
F.7.2 Abstraction definition (BASE) ... 348
F.7.3 Abstraction definition (EXTENDED) ... 354
F.7.4 Abstractor (BASE) ... 365
F.7.5 Abstractor (EXTENDED) ... 366
F.7.6 Access Policy (BASE) ... 370
F.7.7 Access handle (BASE) ... 370
F.7.8 Access handle (EXTENDED) ... 371
F.7.9 Access policy (BASE) ... 372
F.7.10 Access policy (EXTENDED) ... 379
F.7.11 Address space (BASE) ... 386
F.7.12 Address space (EXTENDED) ... 389
F.7.13 Array (BASE) ... 392
F.7.14 Array (EXTENDED) ... 393
F.7.15 Assertion (BASE) ... 395
F.7.16 Assertion (EXTENDED) ... 395
F.7.17 Bus definition (BASE) ... 396
F.7.18 Bus definition (EXTENDED) ... 397
F.7.19 Bus interface (BASE) ... 399
F.7.20 Bus interface (EXTENDED) ... 406
F.7.21 CPU (BASE) ... 413
F.7.22 CPU (EXTENDED) ... 415
F.7.23 Catalog (BASE) ... 417
F.7.24 Catalog (EXTENDED) ... 418
F.7.25 Choice (BASE) ... 421
F.7.26 Choice (EXTENDED) ... 422
F.7.27 Clearbox (BASE) ... 422
F.7.28 Clearbox (EXTENDED) ... 423
F.7.29 Component (BASE) ... 424
F.7.30 Component (EXTENDED) ... 427
F.7.31 Configurable element (BASE) ... 436
F.7.32 Configurable element (EXTENDED) ... 437
F.7.33 Constraint (BASE) ... 438

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

F.7.34 Constraint (EXTENDED) 441
F.7.35 Constraint Set (BASE) 443
F.7.36 Constraint Set (EXTENDED) 444
F.7.37 Design (BASE) 444
F.7.38 Design (EXTENDED) 448
F.7.39 Design configuration (BASE) 454
F.7.40 Design configuration (EXTENDED) 458
F.7.41 Driver (BASE) 461
F.7.42 Driver (EXTENDED) 467
F.7.43 Element attribute (BASE) 470
F.7.44 Element attribute (EXTENDED) 485
F.7.45 File builder (BASE) 508
F.7.46 File builder (EXTENDED) 514
F.7.47 File set (BASE) 520
F.7.48 File set (EXTENDED) 525
F.7.49 Generator (BASE) 531
F.7.50 Generator (EXTENDED) 533
F.7.51 Generator chain (BASE) 535
F.7.52 Generator chain (EXTENDED) 537
F.7.53 Indirect interface (BASE) 539
F.7.54 Indirect interface (EXTENDED) 545
F.7.55 Instantiation (BASE) 551
F.7.56 Instantiation (EXTENDED) 555
F.7.57 Memory map (BASE) 559
F.7.58 Memory map (EXTENDED) 569
F.7.59 Miscellaneous (BASE) 580
F.7.60 Miscellaneous (EXTENDED) 583
F.7.61 Module parameter (BASE) 585
F.7.62 Module parameter (EXTENDED) 586
F.7.63 Name group (BASE) 586
F.7.64 Name group (EXTENDED) 587
F.7.65 Parameter (BASE) 588
F.7.66 Parameter (EXTENDED) 589
F.7.67 Port (BASE) 590
F.7.68 Port (EXTENDED) 604
F.7.69 Port map (BASE) 626
F.7.70 Port map (EXTENDED) 628
F.7.71 Power (BASE) 630
F.7.72 Power (EXTENDED) 631
F.7.73 Register (BASE) 633
F.7.74 Register (EXTENDED) 644
F.7.75 Register file (BASE) 654
F.7.76 Register file (EXTENDED) 656
F.7.77 Slice (BASE) 658
F.7.78 Slice (EXTENDED) 663
F.7.79 Top element (BASE) 667
F.7.80 Top element (EXTENDED) 668
F.7.81 Type definitions (BASE) 670
F.7.82 Type definitions (EXTENDED) 677
F.7.83 Vector (BASE) 686
F.7.84 Vector (EXTENDED) 686
F.7.85 Vendor extensions (BASE) 687
F.7.86 Vendor extensions (EXTENDED) 687
F.7.87 View (BASE) 687

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

F.7.88 View (EXTENDED) 689
F.7.89 All ID types 690

Annex G External bus with an internal/digital interface 693

G.1 Example: Ethernet interfaces 693
G.2 Example: I2C bus 694

Annex H Bridges and channels 695

H.1 Transparent bridge 696
H.2 Opaque bridge 697
H.2.1 Without an address space segment reference 697
H.2.2 With an address space segment reference 698
H.2.3 Effect of an initiator interface address space base address 699
H.3 Channel with address remapping 701
H.4 Channel with bit steering 702

Annex I Examples 705

I.1 abstractionDefinition - RTL 705
I.2 abstractionDefinition - TLM 707
I.3 abstractor 708
I.4 busDefinition 709
I.5 catalog 710
I.6 component 712
I.7 design 736
I.8 designConfiguration 739
I.9 fieldAccessPolicyDefinitions 740
I.10 generatorChain 743
I.11 typeDefinitions 744

Annex J Participants 749

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC document(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

   IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards. Use of an IEEE standard is wholly voluntary.IEEE documents are made available for use subject to important notices and legal disclaimers (see https://standards.ieee.org/ipr/disclaimers.html for more information).

   IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations. This Dual Logo International Standard was jointly developed by the IEC and IEEE under the terms of that agreement.

2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE SA) Standards Board.

3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

IEC 62014-4/ IEEE Std 1685 was processed through IEC technical committee 91: Electronics assembly technology, under the IEC/IEEE Dual Logo Agreement. It is an International Standard.

The text of this International Standard is based on the following documents:

IEEE Std	FDIS	Report on voting
1685 (2022)	91/2025/FDIS	91/2036/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

The IEC Technical Committee and IEEE Technical Committee have decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

IEEE Std 1685™-2022
(Revision of IEEE Std 1685-2014)

IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows

Developed by the

Design Automation Standards Committee
of the
IEEE Computer Society

Approved 21 September 2022

IEEE SA Standards Board

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Abstract: Conformance checks for eXtensible Markup Language (XML) data designed to describe electronic systems are formulated by this standard. The meta-data forms that are standardized include components, systems, bus interfaces and connections, abstractions of those buses, and details of the components including address maps, register and field descriptions, and file set descriptions for use in automating design, verification, documentation, and use flows for electronic systems. A set of XML schemas of the form described by the World Wide Web Consortium (W3C®) and a set of semantic consistency rules (SCRs) are included. A generator interface that is portable across tool environments is provided. The specified combination of methodology-independent meta-data and the tool-independent mechanism for accessing that data provides for portability of design data, design methodologies, and environment implementations.

Keywords: abstraction definitions, address space specification, bus definitions, design environment, EDA, electronic design automation, electronic system level, ESL, IEEE 1685™, implementation constraints, IP-XACT, register transfer level, RTL, SCRs, semantic consistency rules, TGI, tight generator interface, tool and data interoperability, use models, XML design meta-data, XML schema

This publication is dedicated to past editor and friend Joe Daniels, who passed away about 8 months after the team started to work on this revision in Accellera. Your work will live on as will your memory.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1]

Comments on standards should be submitted using the Contact Us form.[2]

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

[1] Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.

[2] Available at: https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; https://www.copyright.com/. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

[3] Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.

[4] Available at: https://standards.ieee.org/standard/index.html.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.[5]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

[5] Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Introduction

This introduction is not part of IEEE Std 1685-2022, IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows.

The purpose of this standard is to provide the electronic design automation (EDA), semiconductor, electronic design intellectual property (IP) provider, and system design communities with a well-defined and unified specification for the meta-data that represents the components and designs within an electronic system. The goals of this specification are to enable delivery of compatible IP descriptions from multiple IP vendors; to improve the importing and exporting of complex IP bundles to, from, and between EDA tools for system on chip (SoC) design environments (DEs); to improve the expression of configurable IP by using IP meta-data; and to improve the provision of EDA vendor-neutral IP creation and configuration scripts (*generators*). The data and data access specification is designed to coexist and enhance the hardware description languages (HDLs) presently used by designers while providing capabilities lacking in those languages.

The SPIRIT Consortium, which originally developed the IP-XACT standard before merging with Accellera, was a consortium of electronic system, IP provider, semiconductor, and EDA companies. IP-XACT enables a productivity boost in design, transfer, validation, documentation, and use of electronic IP and covers components, designs, interfaces, and details thereof. The data specified by IP-XACT is extensible in locations specified in the schema.

IP-XACT enables the use of a unified structure for the meta specification of a design, components, interfaces, documentation, and interconnection of components. This structure can be used as the basis of both manual and automatic methodologies. IP-XACT specifies the tight generator interface (TGI) for access to the data in a vendor-independent manner.

This standardization project provides electronic design engineers with a well-defined standard that meets their requirements in structured design and validation and enables a step function increase in their productivity. This standardization project will also provide the EDA industry with a standard to which they can adhere and that they can support in order to deliver their solutions in this area.

Accellera has prepared a set of bus and abstraction definitions for several common buses. It is expected, over time, that standards groups and manufacturers who define buses will include IP-XACT eXtensible Markup Language (XML) bus and abstraction definitions in their set of deliverables. Until that time, and to cover existing useful buses, a set of bus and abstraction definitions for common buses has been created.

A set of reference bus and abstraction definitions allows many vendors who define IP using these buses to easily interconnect IP together. Accellera posts these definitions for use by its members, with no warranty of suitability, but in the hope that they will be useful. Accellera will, from time to time, update these files and, if a standards body wishes to take over the work of definition, will transfer that work to that body.

These reference bus and abstraction definition templates (with comments and examples) are available from the public area of the Accellera website.[6]

NOTE—Accellera IP-XACT WG Comments on IEEE Std 1685-2014 are acknowledged.

[6]Available at http://www.accellera.org.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows

1. Overview

This clause explains the scope and purpose of this standard; gives an overview of the basic concepts, major semantic components, and conventions used in this standard; and summarizes its contents.

1.1 Scope

This standard describes an eXtensible Markup Language (XML) schema[7] for meta-data documenting *intellectual property* (IP) used in the development, implementation, and verification of electronic systems. This schema provides both a standard method to document IP that is compatible with automated integration techniques and a standard method (generators) for linking tools into a *system development* framework, enabling a more flexible, optimized development environment. Tools compliant with this standard will be able to interpret, configure, integrate, and manipulate IP blocks that comply with the IP meta-data description. The standard is independent of any specific design processes. It does not cover behavioral characteristics of the IP that are not relevant to integration.

1.2 Purpose

This standard enables the creation and exchange of IP in a highly automated design environment.

[7]Information on references can be found in Clause 2.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (shall equals is required to).[8, 9]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (should equals is recommended that).

The word *may* is used to indicate a course of action permissible within the limits of the standard (may equals is permitted to).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (can equals is able to).

1.4 Design environment

The IP-XACT specification is a mechanism to express and exchange information about design IP and its required configuration.[10] While the IP-XACT description formats are the core of this standard, describing the IP-XACT specification in the context of its basic use model, the design environment (DE), more readily depicts the extent and limitations of the semantic intent of the data. The DE coordinates a set of tools and IP, or expressions of that IP (e.g., models), through the creation and maintenance of meta-data descriptions of the system on chip (SoC) so that its system design and implementation flows are efficiently enabled and reuse centric.

The use of the IP-XACT-specified formats and interfaces are shown in **bold** in Figure 1 and described in the following subclauses.

[8]The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

[9]The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

[10]IP-XACT uses the World Wide Web Consortium (W3C®) standard for the XML version 1.0 data (http://www.w3.org/TR/REC-xml/). The valid format of that XML data is described in a *schema* by using the Schema Description Language described therein. W3C is a registered trademark of the World Wide Web Consortium. In addition to XML, many configurability aspects of IP-XACT are inspired from the IEEE 1800 SystemVerilog standard.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

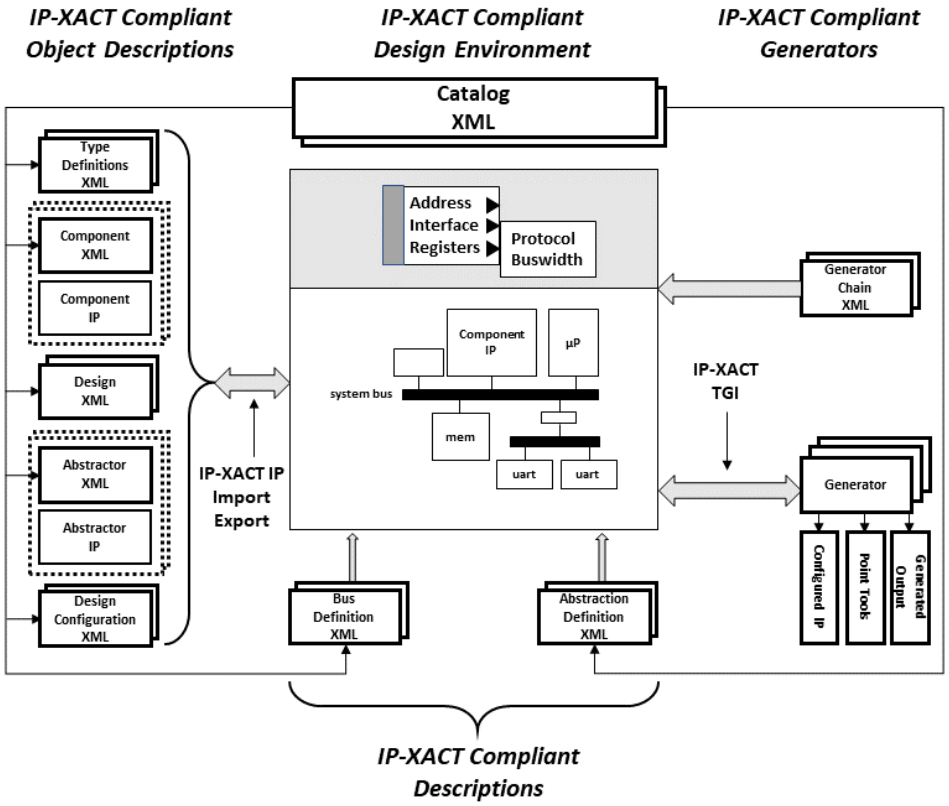


Figure 1—IP-XACT design environment

1.4.1 IP-XACT design environment

A DE enables the designer to work with IP-XACT design IP through a coordinated front-end and IP design database. These tools create and manage the top-level meta-description of system design and may provide two basic types of services: *design capture*, which is the expression of design configuration by the IP provider and design intent by the IP user, and *design build*, which is the creation of a design (or design model) to those intentions.

As part of design capture, a system design tool shall recognize the structure and configuration options of imported IP. In the case of *structure*, this implies both the structure of the design (e.g., how specific pin-outs refer to lines in the hardware description language (HDL) code) as well as the structure of the IP package (e.g., where design descriptions and related generators are provided in the packaged IP data-structure). In the case of *configuration*, this is the set of options for handling the imported IP (e.g., setting the base address and offset, bus width) that may be expressed as configurable parameters in the IP-XACT meta-data.

As part of design build, generators may be provided internally by a system design tool to achieve the required IP integration or configuration, or they may be provided externally (e.g., by an IP provider) and launched by the system design tool as appropriate.

The *system design tool set* defines a DE where the support for conceptual context and management of IP-XACT meta-data resides. However, the IP-XACT specifications make no requirements upon system design

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

tool architecture or a tool's internal data structures. To be considered IP-XACT enabled, a system design tool shall support the import/export of IP expressed with valid IP-XACT meta-data for both component IP and designs, and it needs to support the tight generator interface (TGI) for interfacing with external generators (to the DE).

1.4.2 IP-XACT object descriptions

The IP-XACT schema is the core of the IP-XACT specification. There are nine top-level schema definitions. Each schema definition can be used to create object descriptions of the corresponding types:

- A *bus definition* description defines the type attributes of a bus.
- An *abstraction definition* description defines the representation attributes of a bus.
- A *typeDefinitions* description defines memory map–related elements, including registers, of a component.
- A *component* description defines an IP or interconnect structure.
- A *design* description defines the configuration of and interconnection between components.
- An *abstractor* description defines an adaptor between interfaces of two different abstractions.
- A *generator chain* description defines the grouping and ordering of generators.
- A *design configuration* description defines additional configuration information for a generator chain or design description.
- A *catalog* description provides a mapping between IP-XACT VLNVs (see 1.4.3) and the physical location of the IP-XACT file defining the IP-XACT object with the given VLNV.

1.4.3 Object interactions

An object description contains a unique identifier in the header. The identifier in IP-XACT terms is called a *VLNV* after the four elements that define its value: vendor, library, name, and version. See C.28 for further details on a VLNV. This VLNV is used to create a reference from one description to another. The links between these objects are illustrated in Figure 2. The arrows (A → B) illustrate a reference of one object to another (e.g., reference of object `B` from object `A`).

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

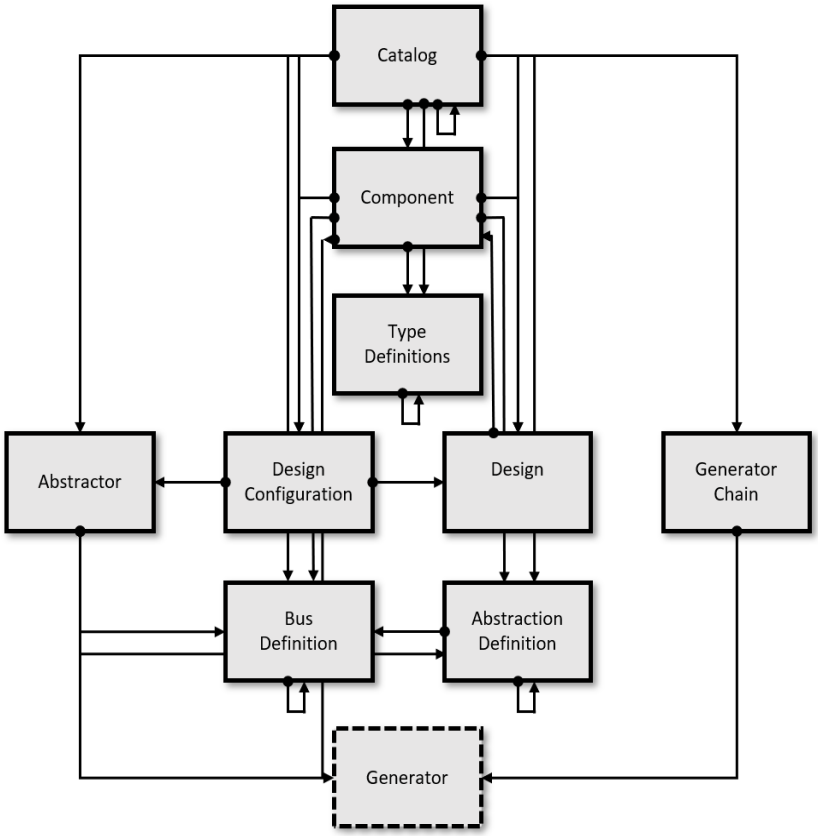


Figure 2—IP-XACT object interactions

1.4.4 IP-XACT generators

Generators are executable objects (e.g., scripts or binary programs) that may be integrated within a DE (referred to as *internal*) or provided separately as an executable (referred to as *external*). Generators may be provided as part of an IP package (e.g., for configurable IP, such as a bus-matrix generator) or as a way of wrapping point tools for interaction with a DE (e.g., an external design netlister, external design checker); see 1.5.2.

An internal generator may perform a wide variety of tasks and may access IP-XACT–compliant meta-data by any method a DE supports. IP-XACT does not describe these protocols.

An *external generator* (often referred to as a *TGI generator*) is an executable program or script invoked from within a DE to query or configure design descriptions and their related component and abstractor descriptions. External generators can use the TGI to *access* their IP-XACT meta-data descriptions (as currently loaded into the DE) and perform the various operations associated with those descriptions. In addition, external generators shall only *operate* upon IP-XACT–compliant meta-data through the defined TGI; see 1.4.6.

Generators can be referenced from a component, abstractor, or generator chain description. Generators can also be grouped and ordered in generator chain descriptions and in chain descriptions contained inside other chain descriptions. This sequencing of generators is *critical* for providing script-based support for SoC flow creation.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

1.4.5 IP-XACT design environment interfaces

There are two interfaces expressed in Figure 1: from the DE to the external IP libraries and from the DE to the generators. In the former case, the IP-XACT specifications are *neutral* regarding the design tool interfaces to IP repositories. Being able to read and write IP with IP-XACT meta-data is required; however, the *formal interaction* between an external IP repository and a DE is not specified. In the latter case, the interface between the DE and a generator is the TGI; see 1.4.6.

1.4.6 Tight generator interface

The *tight generator interface* (TGI) is the method a generator uses to access a design or component description in a DE-independent and generator-language-independent manner. Therefore, a generator running on two different DEs produces the same results. The DE and the generator can communicate with each other by sending messages utilizing the Simple Object Access Protocol (SOAP)[11] specified in the Web Services Description Language (WSDL).[12] SOAP or REST[13] provide a simple means for sending XML-format messages using the Hypertext Transfer Protocol (HTTP) or other transport protocols. IP-XACT supports using HTTP or a file protocol.

The messages passed between the generator and the DE allow the generator to get all information about the design interconnections (which contain components and abstractors), provide set information for any configurable elements in a component or abstractor, and make simple modifications of the design description. For additional details on the DE generator invocation and the messages passed between the generator and the DE, see Annex F.

1.4.7 Design intellectual property

IP-XACT is structured around the concept of IP reuse. *Electronic design intellectual property*, or IP, is a term used in the electronic design automation (EDA) community to refer to a reusable collection of design specifications that represent the behavior, properties, and/or description of the design in various media. The name IP is partially derived from the common practice of considering a collection of this type to be the intellectual property of one party. Both hardware and software collections are encompassed by this term.

These collections may include the following:

a) Design objects—This collection can include the following:
   1) Transaction-level modeling (TLM) descriptions: SystemC® and SystemVerilog [14]
   2) Fixed HDL descriptions: Verilog®, VHDL[15]
   3) Configurable HDL descriptions (e.g., bus-fabric generators)
   4) Design models for register transfer level (RTL) and transactional simulation (e.g., compiled core models)
   5) HDL-specified verification IP (VIP) (e.g., basic stimulus generators and checkers)

b) IP views—This collection is a list of different views (levels of description and/or languages) to describe the IP object. In IP-XACT, these views include the following:
   1) Design view: RTL Verilog or VHDL, flat or hierarchical components
   2) Simulation view: model views, targets, simulation directives, etc.
   3) Documentation view: standard, user guide, etc.

[11]Available from the W3C Web site at http://www.w3.org/TR/soap12-part1/.
[12]Available from the W3C Web site at http://www.w3.org/TR/wsdl.
[13]Available from the W3C Web site at https://www.w3.org/TR/ws-arch/#relwwwrest
[14]SystemC is a registered trademark of the Accellera Systems Initiative.
[15]Verilog is an abandoned trademark.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

IP-XACT XML meta-data descriptions provide a standardized way of collecting much of the structural information contained in the file sets. IP-XACT also can contain the information that identifies the appropriate files included in a collection to be used for different parts of the design process.

1.5 IP-XACT–enabled implementations

Complying with the rules outlined in this subclause allows providers of tools, IP, or generators to class their products as *IP-XACT enabled*. Conversely, any violation of these rules removes that naming right. This subclause first introduces the set of metrics for measuring the valid use of the specifications. It then specifies when those validity checks are performed by the various classes of products and providers: DEs, point tools, IPs, and generators.

a) Parse validity
   1) Parsing correctness: Ability to read all IP-XACT descriptions.
   2) Parsing completeness: Cannot require information that could be expressed in an IP-XACT format to be specified in a non-IP-XACT format. Processing of all information present in an IP-XACT document is not required.

b) Description validity
   1) Schema correctness: IP is described using XML files that conform to the IP-XACT schema.
   2) Usage completeness: Extensions to the IP-XACT schema shall be used only to express information that cannot otherwise be described in IP-XACT.

c) Semantic validity
   1) Semantic correctness: Adheres to the semantic interpretations of IP-XACT data described in this standard.
   2) Semantic completeness: Obeys all the semantic consistency rules (SCRs) described in Annex B.

These validity rules can be combined with the product class-specific rules to cover the full IP-XACT–enabled space. The following subclauses describe the rules a provider has to check to claim a product is IP-XACT enabled.

An IP-XACT–enabled DE or point tool may read descriptions based on multiple versions of the IP-XACT schema. If the DE or point tool does provide this capability, the effect shall be as if all of the descriptions had been translated to the highest schema version supported by the given tool. This is required for semantic consistency. Schema version translation can be done in a number of different ways, but the most common is to leverage the eXtensible Stylesheet Language Transformations (XSLT)[16] provided with the IP-XACT schema image. In addition, a DE or point tool may preserve information in the initial description for use outside of the scope of the IP-XACT specification.

1.5.1 Design environments

An IP-XACT–enabled DE shall

- Follow the parse validity requirements shown in 1.5.
- Create IP that is IP-XACT enabled.
- Modify any existing IP-XACT descriptions without losing any preexisting information. In particular, it shall preserve any vendor extension data included in the existing IP-XACT description.
- Be able to invoke IP-XACT–enabled generators with **apiType** of **none** (see 6.16.2).

[16] Available from the W3C Web site at http://www.w3.org/TR/xslt.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

An IP-XACT–TGI–enabled DE shall be considered as enabled as follows:

— For the base TGI, if it supports all generators utilizing the `TGI_2022_BASE` **apiType** (see 6.16.2).
— For the extended TGI, if it supports all generators utilizing the `TGI_2022_BASE` or `TGI_2022_EXTENDED` **apiType**s (see 6.16.2).

1.5.2 Point tools

A point tool is a tool that has a particular, rather than a general, set of capabilities. In contrast to an IP-XACT–enabled DE (see 1.5.1), an IP-XACT–enabled point tool shall

— Follow the parse validity requirements shown in 1.5.
— Create IP that is IP-XACT enabled.
— Modify any existing IP-XACT descriptions whiletrying not to lose any preexisting information. In particular, it shall preserve any vendor extension data included in the existing IP-XACT description.

1.5.3 IPs

An IP-XACT–enabled IP shall

— Have an IP-XACT description that follows the description and semantic validity requirements shown in 1.5.
— Use IP-XACT–enabled generators for any generators associated with this IP.

XML descriptions compliant with IP-XACT shall provide a namespace reference to the `index.xsd` schema file, not to any of the other files in the release.

1.5.4 Generators

An IP-XACT–enabled generator shall

— Create IP that is IP-XACT enabled.
— Modify any existing IP-XACT descriptions without losing any preexisting information. In particular, it shall preserve any vendor extension data included in the existing IP-XACT description.
— Communicate with the DE that invoked it only through the IP-XACT TGI (see Annex F).

1.6 Conventions used

The conventions used throughout the document are included here.

IP-XACT is case-sensitive.

1.6.1 Visual cues (meta-syntax)

Bold shows required keywords and/or special characters, e.g., **addressSpace**. For the initial definitional use (per element), keywords are shown in **boldface-red text**, e.g, **bitsInLau** (see also 1.7).

Bold italics shows group names or data types, e.g., ***nameGroup*** or ***boolean***. For definitions of types, see Annex D.

`Courier` shows examples, external command names, directories and files, etc., e.g., address `0x0` is on `D[31:0]`.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

1.6.2 Notational conventions

The keywords *required, shall, shall not, should, should not, recommended, may,* and *optional* in this standard are to be interpreted as described in 1.3 and IETF RFC-2119 [B5].[17] When a conflict between the two exists, the use in 1.3 takes precedence.

1.6.3 Syntax examples

Any syntax examples shown in this standard are for information only and are intended only to illustrate the use of such syntax (see also Annex I).

1.6.4 Graphics used to document the schema

The W3C Web site[18] specifies the XML schema language used to define the IP-XACT XML schemas. Normative details for compliance to the IP-XACT standard are contained in the schema files. Within this document, pictorial representations of the information in the schema files *illustrate* the structure of the schema and *define* any constraints of the standard. With the exception of visibility issues, the information in the figures and the schema files is intended to be identical (for a fully annotated version of the schema files; see IP-XACT Schema [B8]). Where the figures and schema are in conflict, the XML schema file shall take precedence.[19]

1.6.4.1 Elements and attributes

The *element* is the fundamental building block on which this standard is based. An element may be either a *leaf element*, which is a container for information, or a *branch element*, which may contain further branch elements or leaf elements.

A leaf or branch element may also contain *attributes*. Attributes are containers for information within the containing element.

1.6.4.2 Types

A *type* is a designation of the format for the contents of an element or attribute. There are two different styles of types that can be defined. A type may be assigned to a leaf element or an attribute. This type is called a ***simpleType*** and defines the format of data that may be stored in this container. A type may also be assigned to a branch element. This type is called a ***complexType*** and defines further elements and attributes contained in the branch element.

1.6.4.3 Groups

A group is a collection of elements or attributes, which allow the same collection of items to be referenced consistently in many places. There are two different types of groups that can be defined. A *group* is a combination of leaf or branch elements; an ***attributeGroup***, a simple list of attributes. The names assigned to either group have no representation in the resulting description.

[17]The number in brackets correspond to the numbers of the bibliography in Annex A.

[18]Available from the W3C Web site at https://www.w3.org/TR/REC-xml/.

[19]The graphics for this document have been generated by taking “screen-shots” of the various files as they are displayed in Altova’s XML environment XMLSpy®. XMLSpy is a registered trademark of Altova GmbH. This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of this product. Equivalent products may be used if they can be shown to lead to the same results.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

1.6.4.4 Namespace

Each element, attribute, type, or group has a name, which is preceded by a namespace-prefix and separated from the name by a colon (**:**). For the examples in 1.6.4.5, **xyz** is used as the namespace-prefix for all of the items whereas this standard uses **ipxact**. Within the text of this standard, the namespace-prefix is not written when describing an item; it is shown only in examples.

1.6.4.5 Diagrams

The diagrams used throughout this standard graphically detail the organization of elements and attributes.

NOTE—For brevity, the **xml:id** attribute (see C.30) has been removed from all diagrams.[20]

1.6.4.5.1 Elements and sequences

Figure 3 shows the sequence-compositor. At the left is a branch element, **element1**, with some descriptive text below. **element1** is connected to a sequence-compositor. The sequence-compositor defines the order the subelements appear in the branch element. **subElement1** shall appear first inside of **element1**. This is followed by **subElement2**, **subElement3**, **subElement4**, and **subElement5** before closing **element1**.

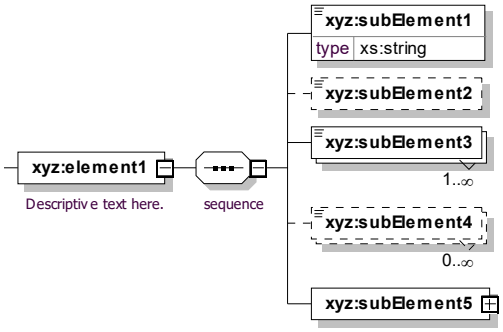


Figure 3—Sequence-compositor

a) **subElement1** is a mandatory element, as indicated by the solid line of the containing box. The type of the data contained in this element is set to ***string***, and it has a default value of `ipxact` if the element is present but left empty.

b) **subElement2** is an optional element, as indicated by the dashed-line of the containing box.

c) **subElement3** is an mandatory element that may appear multiple times, indicated by the double-solid line of the containing box. The number of times the element may appear is indicated by the range of the numbers listed below the element.

d) **subElement4** is an optional element that may appear multiple times, as indicated by the double-dashed line of the containing box. The number of times the element may appear is indicated by the range of the numbers listed below the element.

e) **subElement5** is an mandatory branch element that contains further elements inside, as indicated by the small plus sign (+) in the small box on the right.

[20]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

Figure 4 shows variations of a sequence-compositor. **root1** is connected to an optional sequence-compositor, as indicated by the symbol being drawn with a dashed line. **element1** may appear first inside of **root1**; if it does, it shall be followed by **element2**. Each subelement is connected to a sequence-compositor.

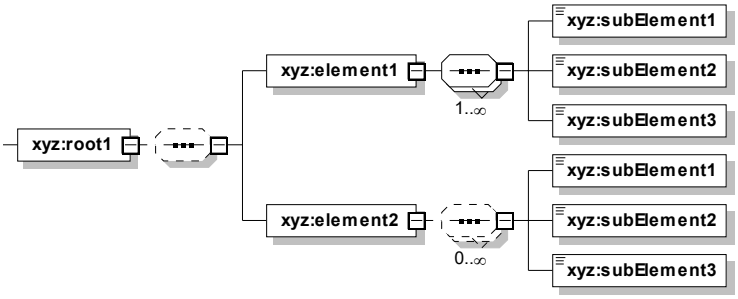


Figure 4—Sequence-compositor variations

— **element1** may contain one or more of the following sequences in the following order: **subElement1** and **subElement2** and **subElement3**. The number of times the sequence-compositor may appear is indicated by the range of the numbers listed below the symbol. If the range is greater than 1, the sequence-compositor symbol is drawn with double lines.

— **element2** is optional and may contain zero or more of the following sequences in the following order: **subElement1** and **subElement2** and **subElement3**. The number of times the sequence-compositor may appear is indicated by the range of the numbers listed below the symbol. If the range starts at 0 and the maximum is greater than 1, the sequence-compositor symbol is drawn with double-dashed lines.

1.6.4.5.2 Elements and choices

Figure 5 shows the variations of the choice-compositor. **root** is connected to a choice-compositor. The choice-compositor specifies that one of the elements on the right side shall be chosen. **root** may contain one of the following: **element1**, **element2**, or **element3**. Each subelement is connected to a choice-compositor.

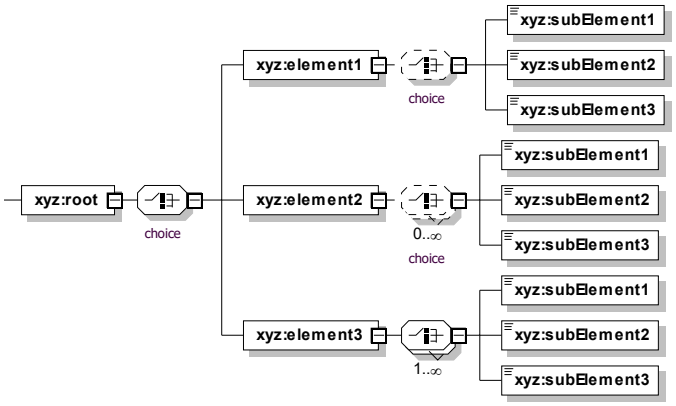


Figure 5—Choice-compositor variations

a) **element1** may contain one of the following: **subElement1**, **subElement2**, or **subElement3**, as indicated by the symbol being drawn with a dashed line.

b) **element2** may contain any (0 or more) of the following: **subElement1**, **subElement2**, or **subElement3** in any order. The number of times the choice-compositor may appear is indicated by

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

the range of the numbers listed below the symbol. If the range starts at 0, the choice-compositor is drawn with dashed lines.

c) **element3** may contain one or more of the following: **subElement1**, **subElement2**, or **subElement3** in any order. The number of times the choice-compositor may appear is indicated by the range of the numbers listed below the symbol. If the range is greater than 1, the choice-compositor is drawn with double lines.

1.6.4.5.3 Elements, attributes, groups, and attributeGroups

Figure 6 shows the use of attributes, groups, and attributeGroups. **element1** contains two attributes, shown in the tab-shaped box labeled *attributes*. **attribute1** is optional, as indicated by the dashed containing box, and is of type ***integer*** with a default value of 7 if the attribute is not present. **attribute2** is a required attribute, as indicated by the solid containing box, and is of type ***boolean*** with no default. The ordering in which **attribute1** and **attribute2** appear inside **element1** is irrelevant.

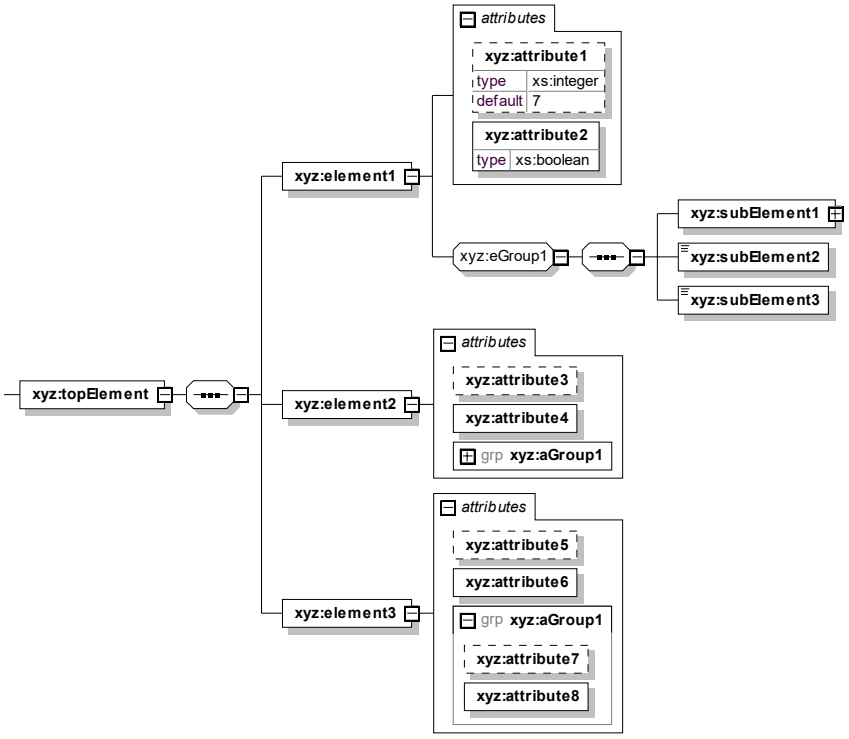


Figure 6—Attributes, groups, and attributeGroups

a) ***eGroup1*** is an element group inside **element1**. This group contains three subelements, and the group symbol can be replaced by a solid line. The name of the group has no representation in the resulting output description. An element group can be optional, as indicated by a dashed outline (not shown), and it can also have a range, as indicated by numbers below the group symbol (not shown).

b) ***aGroup1*** is an ***attributeGroup*** inside **element2** and **element3**. This ***attributeGroup*** contains two attributes, **attribute7** and **attribute8**. Inside **element2**, the ***attributeGroup*** is shown in its collapsed form, as indicated by the small plus sign (+) inside the small box. Inside **element3**, the ***attributeGroup*** is shown in its expanded form, as indicated by the small minus sign (−) inside the small box. **element2** contains four attributes: **attribute3**, **attribute4**, **attribute7**, and **attribute8**. **element3** also contains four attributes: **attribute5**, **attribute6**, **attribute7**, and **attribute8**. The name of the ***attributeGroup*** has no representation in the resulting description.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

1.6.4.5.4 Wildcards

Figure 7 shows the use of wildcards. A *wildcard* is depicted by the rounded box with the **any ##any** text. Wildcards indicate that any well-formed attribute or element may be inserted into the containing element.

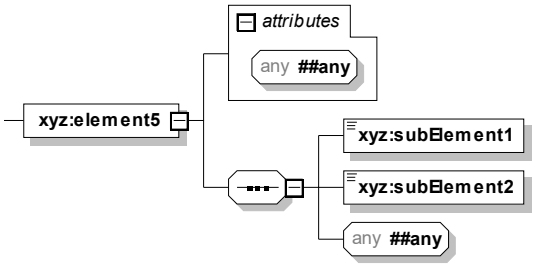


Figure 7—Wildcards

1.7 Use of color in this standard

This standard uses a minimal amount of color to enhance readability. The coloring is not essential and does not affect the accuracy of this standard when viewed in pure black and white. The places where color is used are the following:

- Cross references that are hyperlinked to other portions of this standard are shown in underlined-blue text (hyperlinking works when this standard is viewed interactively as a PDF file).
- In the formal language definitions, syntactic keywords and tokens are shown in **boldface-red text**.

1.8 Contents of this standard

The organization of the remainder of this standard is as follows:

- Clause 2 provides references to other applicable standards that are assumed or required for this standard.
- Clause 3 defines terms, acronyms, and abbreviations used throughout the different specifications contained in this standard.
- Clause 4 defines the interoperability use model.
- Clause 5 defines the bus and abstraction interface definitions.
- Clause 6 defines the component and interconnect models.
- Clause 7 defines the designs and their connections.
- Clause 8 defines the abstractor model between abstraction definitions.
- Clause 9 defines the type definition descriptions.
- Clause 10 defines the generator chain configuration.
- Clause 11 defines design configurations.
- Clause 12 defines the catalog feature.
- Clause 13 defines addressing.
- Clause 14 defines data visibility.
- Annex A Bibliography.
- Annex B Semantic consistency rules.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

— Annex C Common elements and concepts.
— Annex D Types.
— Annex E SystemVerilog expressions.
— Annex F Tight generator interface.
— Annex G External bus with an internal/digital interface.
— Annex H Bridges and channels.
— Annex I Examples.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used; therefore, each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

eXtensible Markup Language (XML) schema specification, available from the W3C Web site at http://www.w3.org/TR/xmlschema-0/; http://www.w3.org/TR/xmlschema-1/; http://www.w3.org/TR/xmlschema-2/.

eXtensible Markup Language (XML) version 1.0 specification, available from the W3C Web site at http://www.w3.org/TR/REC-xml/.

eXtensible Stylesheet Language Transformations (XSLT) version 1.0 specification, available from the W3C Web site at http://www.w3.org/TR/xslt.

IEEE Std 1800™, IEEE Standard for SystemVerilog-Unified Hardware Design, Specification, and Verification Language.[21, 22]

REpresentational State Transfer (REST), Web Services Architecture, description of relationship to World Wide Web on the W3C Web site at https://www.w3.org/TR/ws-arch/#relwwwrest

Simple Object Access Protocol (SOAP) version 1.2 specification, available from the W3C Web site at http://www.w3.org/TR/soap12-part1/.

Web Services Description Language (WSDL) version 1.1 specification, available from the W3C Web site at http://www.w3.org/TR/wsdl.

XML schema NameChar definition, available from the W3C Web site at http://www.w3.org/TR/REC-xml/#NT-NameChar.

XML schema NameStartChar definition, available from the W3C Web site at http://www.w3.org/TR/REC-xml/#NT-NameStartChar.

[21]IEEE publications are available from The Institute of Electrical and Electronics Engineers, Inc. (http://standards.ieee.org/).
[22]The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

Published by IEC under licence from IEEE. © 2022 IEEE. All rights reserved.