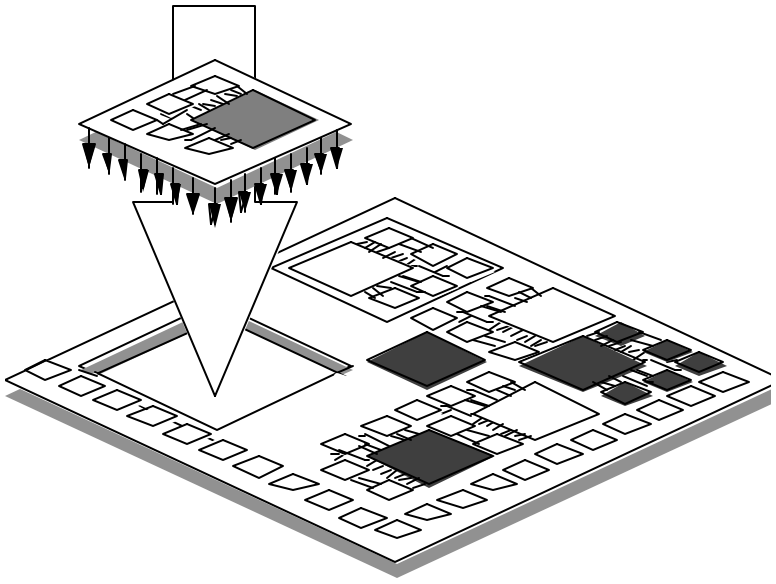
VSI Alliance™
Virtual Component Attributes (VCA)
With Formats for Profiling, Selection, and Transfer
Standard Version 2.3
(VCT 2 2.3)

Virtual Component Transfer Development Working Group

March 2003



Dedication to Public Domain

VSI Alliance hereby dedicates all copyright that VSI Alliance holds in this _____ (the "Work") to the public domain, free of charge, and for the general benefit of the public at large.

VSI Alliance intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights that VSI Alliance may have in the Work under copyright law, whether vested or contingent, including without limitation, the right to prevent others from freely reproducing, distributing, transmitting, using, modifying, building upon or otherwise exploiting the Work for any purpose, commercial or non-commercial, or in any way.

VSI Alliance understands that such relinquishment includes the relinquishment of all rights to enforce (by lawsuit or otherwise) any copyrights that VSI Alliance may have in the Work.

IMPORTANT - NO WARRANTY. THE WORK IS PROVIDED "AS IS", "WHERE-IS", WITHOUT WARRANTY OF ANY KIND. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, VSI ALLIANCE EXPRESSLY DISCLAIMS ALL WARRANTIES WITH RESPECT TO THE WORK, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, WARRANTIES OF TITLE, NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, AND IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Virtual Component Transfer Development Working Group

Company Members of the Development Working Group:

Alcatel
ARM
Cadence Design Systems
Design & Reuse
Fujitsu
Hewlett-Packard
Infineon Technologies
NOKIA Mobile Phones
Oki Electronics
Silicon Integration Initiative
STMicroelectronics
Synchronicity
Virtual Component Exchange (VCX)

Individual Members of the Development Working Group:

Lee Dilley

Contributors:

Patrick Cochennec . . . . . STMicroelectronics
Lee Dilley . . . . . Individual Member
Takeshi Fuse (Prior Chairman) . . . . . Fujitsu Ltd.
Katherine Hsu . . . . . Synchronicity
Takahide Inoue . . . . . VSIA Staff
Jukka.T.Liimatainen . . . . . NOKIA
William T. McVay, Jr. (Chairman) . . . . . Alcatel
Caroline O'Donnell VCX
Ian Phillips . . . . . ARM Ltd.
Gordon Price (Vice-Chairman) VCX
Gabriele Saucier, Ranto Rafidinitrimo . . . . . Design & Reuse
John Teets, Melanie Yunk . . . . . Silicon Integration Initiative
Kumar Venkatramani . . . . . Cadence Design Systems
Mobashar Yazdani . . . . . Hewlett-Packard

Technical Editor:

Lee Dilley

Revision History

28Jan00	Version 1.0	Editorial Staff	Inserted DWG member page and edits from TC
31Mar00	Version 1.5	DWG Tiger Team	Added attribute formats
10Apr00	Version 1.9	Lee Dilley	DWG member review released
5May00	Version 1.93	Lee Dilley	Inserted changes from DWG member review
15May00	Version 1.94a	Lee Dilley	TC released
22May00	Version 2.0	Editorial Staff	Formatted for member review, copy edited
09Oct00	Version 2.14	Lee Dilley	Inserted changes from DWG member review
19Oct00	Version 2.15	Lee Dilley	Prepared for Technical Committee
08Nov00	Version 2.18	Lee Dilley	Added descriptor section numbers and repaired formatting
16Nov00	Version 2.2	Lee Dilley	Rewrote Section 2.1 and fixed all internal references
21Nov00	Version 2.2	Editorial Staff	Formated legends and cover for Board Review
12Jan01	Version 2.2	Editorial Staff	Copy edited and formatted
15Jan01	Version 2.2	Herb Leeds	Reviewed and edited
27Feb01	Version 2.2	Editorial Staff	Copy edited and formatted, corrected figures
28Mar03	Version 2.3	Editorial Staff	Corrected typo in table on page 61

Table of Contents

1. Overview

1.1 VCT Documents Organization

The VSI Alliance™ (VSIA) Virtual Component Transfer (VCT) documents define the documentation and design data of virtual components (VCs) so that VCs can be profiled, selected, and transferred between VC providers and VC users in an organized and coherent manner. They are intended to serve as the common technical foundation necessary for the emergence of a viable and vibrant VC exchange marketplace. The three VCT documents are:

- *VC Documentation Requirements* – VCT 1 (first published in October, 1999)
- *VC Attributes with Formats* – VCT 2 (this document)
- *Packaging VC for Transfer* – VCT 3 (targeted for initial release in mid-2001)

VCT 1 (*VSI Alliance Virtual Component Transfer Specification*) is a specification that documents all the information necessary to thoroughly and accurately characterize the VC, and prepares the foundation for VC search, selection, and exchange. VC providers and users are guided through the process of VC integration.

VCT 2 *(VSI Alliance Virtual Component Attributes With Formats for Profiling, Selection, and Transfer Standard)* sets forth the standard for documenting VC exchange by distilling the information identified in VCT 1 into a set of detailed attribute definitions designed to facilitate the search and selection process within the VC integration cycle. The VCT 2 standard provides the attribute structure and syntax needed to support cross-domain (inter-company, intra-company) exchange of catalog information through the Internet and other electronic media such as CD-ROM. VC users will search for the VC in various information databases (VC catalogs) and conduct first-cut comparisons between different offerings utilizing the attributes defined therein. Broad industry acceptance of VCT 2 data types and formats will facilitate the searching of VC catalogs from different vendors using common formats.

VCT 3 will address the problem of how to package and transfer VC design data and documentation as a whole (all VSIA-specified deliverables), so that VCs can be seamlessly exchanged and VCs from different sources can be efficiently mixed and matched in a single system-on-chip design.

1.2 Scope

This document defines a set of attributes for the high-level profiling and characterization of VCs. Attribute definitions include detailed data type information and have a consistent structure. They are designed to facilitate the rapid and easy identification of desired VCs from catalogs and databases containing a large number and variety of VCs. Users depend on such catalog data when they search and select appropriate VCs for their applications. Figure 1: VC Profiling Information Flow, shows the general flow and usage of catalog information in conjunction with this process.

While it is neither practical nor sensible to standardize the formats and contents of proprietary catalog databases, the efficient and unambiguous exchange of catalog data requires definition and compliance with an industry accepted common format. VCT 2 achieves this objective by establishing a standard classification scheme for VC characteristics. The VCT 2 standard imposes no requirements on the formats and structure used within any specific database. To meet the VCT 2 standard, VC catalog providers need merely parse their proprietary formats into this common exchange format and include the documents and design data specified in Table 1 of the *VSI Alliance Deliverables Document* (See Section 1.7; Reference 1 of this document.)

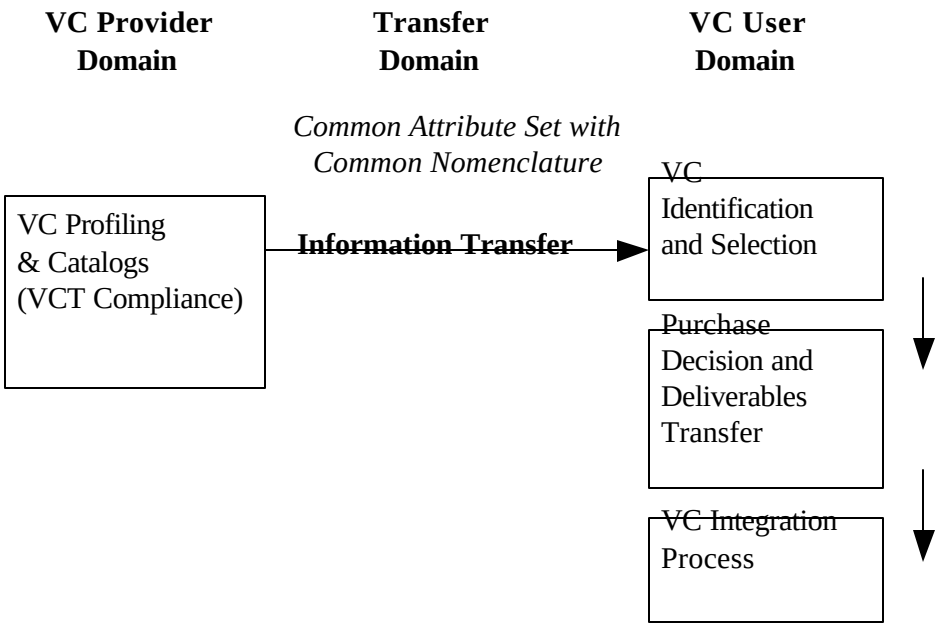


Figure 1: VC Profiling Information Flow

1.3 Referenced Intellectual Property (IP)

This document contains no specifically referenced intellectual property (IP).

1.4 Definition of Terms

VC: Virtual component

VCA: Virtual component attributes

VCT: Virtual component transfer

VC Provider: A person or entity (also known as the VC creator) who originates and sources the VC in the VC transfer process.

VC User: A person or entity (also known as the VC integrator) who receives a VC in the transfer process; the counterpart to the VC provider.

1.5 Methodology Description

The goal of enabling VC providers to produce a single set of VC documentation applicable to all phases of the VC selection, transfer, and integration process is achieved by defining a set of descriptors, attributes, and their value types. This documentation set is suitable for a variety of user needs while satisfying the requirements of the information supplier community for ready access to VC data from a broad range of sources. This set is based on the VC documentation standards first published in *VSI Alliance Virtual Component Transfer Specification (S*ee Section 1.7; Reference 2 of this document) and establishes the baseline criteria for an unambiguous understanding of the VCT 1 terminology.

This document presents VC characterization data derived from each area covered in VCT 1 (for example, Claims, System/Logic Description, and Physical Description). It organizes the data by descriptor category into a consistent set of attributes, which facilitates success in the initial assessment stage (search and select), and is conducive to automation. Attributes have been chosen that can be clearly and unambiguously represented using succinct text strings or numerical expressions. An enumerated list of acceptable attribute field values (a pick list) is included whenever possible to improve consistency and accuracy of VC descriptions.

The VCT 2 standard contains four sections:

- Section 1 (this section) is an introduction to the document. It provides the overview, scope, and details of the methodology used.
- Section 2 establishes the common terminology, nomenclature, and format for profiling VCs by describing the semantics and syntax of each attribute listed. Each attribute is presented with its allowable data types, organized by attribute field. Each attribute field has exactly one value type defined. Provision is made for expanding the pre-defined attribute set by allowing the definition of custom attributes, if needed.
- Section 3 offers general guidelines for the implementation and use of this standard.
- Section 4 is a glossary of acronyms.

Following these sections, five appendices have been included. The appendices provide additional detail about the functional and market segment classification schemes used; organize attribute information; show related pick lists in tabular form; detail attribute data type information in a format useful for implementers; and reference the ISO 6093 Data Type Definitions used in Section 2.

1.6 Summary of Attributes

Descriptors are organized in categories and follow the presentation sequence of the VCT 1 specification. Table 1 indicates each descriptor category by highlighting it on a single line in bold text. Every attribute listed in Table 1 is presented in complete detail in Section 2.2 of this document. For cross-reference purposes, VCT 1 section numbers are provided in the rightmost column. The attributes columns show the component attributes that make up each descriptor. Descriptors that have no applicable attributes are indicated with “N/A” in the attributes column and have no entry in Section 2.2 of this document.

If specific information is available, the VC provider is encouraged to enter it in the attribute field of the most closely matching attribute. If the information fits the attribute fields of more than one attribute, it may be entered into each. Having the same information available under multiple attributes facilitates successful searching by a wide variety of criteria.

Table 1: Summary of Attributes

Section Number	Descriptor Category Description Name VC Provider	Attributes				VCT 1 Sec. #
	Claims					2.1
2.2.1	Functional Overview	Class	Architecture	ModesOf-Operation	Equivalent-PartsOrCores	2.1.1
2.2.2	Target Applications	MarketSegments				2.1.2
2.2.3	Performance	Frequency	PowerConsumption	Throughput	Latency	2.1.3
		Hardness	GateCount	BitCount	TotalArea	2.1.4
2.2.4	Form Information	LogicArea	MemoryArea	PinCount	FlipFlop-Count	
2.2.5	Test Coverage	StuckFault-Coverage	CodeCoverage			2.1.5
2.2.6	List of Deliverables	Deliverables	Availability	Reference		2.1.6
2.2.7	Features and Standards Compliance	Compliant-Standard	Features			2.1.7
	System/Logic Description					2.2
	Functional Description	N/A				2.2.1
2.2.8	Abstract Models	Modeling-Abstractions				2.2.2
2.2.9	Structural Views	Interface-Structure	InternalObserv-abilityStructure			2.2.3

Table 1: Summary of Attributes (Continued)

Section Number	Descriptor Category Description Name	Attributes				VCT 1 Sec. #
2.2.10	Interfaces	PortDescriptions	Ports	Configura-tion-Registers	Interrupt-Interface	2.2.4.1
2.2.11	Logical (Mapped) Interfaces	LogicInterface-Type				2.2.4.2
		Arbitration-Interface	MemoryAllocation			
2.2.12	Integration Requirements	APISupport				2.2.5
2.2.13	System/Logic Test Suite	TestSuites				2.2.6
	Physical Description					2.3
2.2.14	Timing Specification	Duty	Period	Phase		2.3.1.1
2.2.15	Electrical Characteristics	SupplyVoltage	Operating-Temperature	SNR	Sensitivity	2.3.1.2
		I/O-Characteristic	OutputCharacteristic			
	Integration Requirements					2.3.2
2.2.16	Clock Distribution	Clocks				2.3.2.1
2.2.17	Design Constraints	Abstract	Reset			2.3.2.2
	Design Compatibility	N/A				2.3.2.3
	Process Compatibility	N/A				2.3.2.4
2.2.18	Process Requirements	Specified-Processes	SpecifiedFoundries			2.3.2.5
2.2.19	Design Process Sensitivities	Process-Sensitivity				2.3.2.6
2.2.20	Implementation Test Suite	Implementation-TestSuite				2.3.3
	Reference Environment					2.4
2.2.21	Verification of Claims	Qualification-Levels	Certifications			2.4.1
2.2.22	Tools, Flows, and Methodology	EDAtools	SupportTools	Reference-Designs		2.4.2
2.2.23	ASIC Libraries	Library				2.4.3

Table 1: Summary of Attributes (Continued)

Section Number	Descriptor Category Description Name	Attributes				VCT 1 Sec. #
2.2.24	Process Technology	ProvenProcesses	ProvenFoundries			2.4.4
	Naming Convention	N/A				2.4.5
	Deliverables Documentation	N/A				2.4.6
	Application Information					2.5
2.2.25	Version History	Version				2.5.1
2.2.26	Known Bugs	BugList				2.5.2
2.2.27	Application Notes	RelatedVCs	Customization-Options			2.5.3
	Test Information					2.6
2.2.28	Test Strategy	TestMethods				2.6.1
2.2.29	Test Modules	TestModule				2.6.2
	Test Modes	N/A				2.6.3
	Mixed Signal Test Integration	N/A				2.6.4
	Supplier Information					2.7
2.2.30	VC Provider Contact Information	Contact-Information				2.7.1
2.2.31	Transfer Package Information	Package	PackageInformation			2.7.2
2.2.32	Standard Terms and Conditions	SupportModels	FeeModels	Applicable-Patents		2.7.3
2.2.33	Third-Party Reference	Audits	Supporting-VCPartners	ISVs		2.7.4
	Other					N/A
2.2.34	Customization	Custom				N/A

1.7 References

The following released specifications are referred to in this document. The primary VSIA document for referencing all other VSIA specification documents is the *VSI Alliance Deliverables Document* [Reference 1] which contains the most current version numbers. The version consulted for each specification referenced below was determined from the *VSI Alliance Deliverables Document* **Version 2.3** - Revision 17Oct00:

- Reference 1: *VSI Alliance Deliverables Document (DD x.y.)*
- Reference 2: *VSI Alliance Virtual Component Transfer Specification (VCT 1 2.0)*
- Reference 3: *VSI Alliance On-Chip Bus Attributes Specification (OCB 1 1.0)*
- Reference 4: *VSI Alliance System-Level Interface Behavioral Documentation Standard (SLD 1 1.0)*
- Reference 5: *VSI Alliance System-Level Design Model Taxonomy (SLDTAX 1.0)*

1.8 Known Problems and Workarounds

The following known problems exist in this document:

- There is currently no attribute to record rules checking and signoff criteria in Section 2.2.21. It is anticipated that guidance from the VSIA Quality Study Group will provide the appropriate basis for adding such an attribute.
- There is no attribute for route-over constraints in Section 2.2.17.
- There is no attribute for test assumptions; it is recommended that such information be handled by adding remarks to attribute fields wherever deemed appropriate.
- There is no attribute for power structure (metal usage rules, ground constraints, and so forth). This will be added to Section 2.2.17 “Design Constraints” in a future release. For now, utilize remarks in “Usage Notes” in Section 2.2.3.2 “Power Consumption.”

1.9 Planned Future Releases

The VCT DWG intends to review this document for compatibility with other VSIA specifications and standards and to perform routine updates and maintenance. The following are possible enhancements being considered for inclusion in a future release:

- There is currently no attribute to declare the quality measure of the VC. Based on input from the VSIA Quality Study Group, this omission would most likely be corrected by adding an appropriate attribute in Section 2.2.21 (Verification of Claims.)
- Signal Integrity attributes will be considered pending input from the Signal Integrity work of the Analog/ Mixed-Signal Development Working Group (AMS DWG).
- System-to-Logical interface map.

2. VC Attribute Descriptions

2.1 Attribute Structure and Syntax

This section sets forth the syntax used for each attribute listed in Section 2.2 for VC classification. Attributes are organized by category of descriptor, and each descriptor category is referenced to the VCT 1 section where it was first introduced.

2.1.1 Attribute Structure

2.2.x {VCT 1 Section Name}: {Descriptor}

VCT ~~1 Section {Section Number}~~

{Descriptor Explanation}

2.2.x.y {Attribute Name}

{Attribute Description}{compound type if indicated in description here}

{Attribute Field Name}
{Attribute Field Description}

Value {single or multiple type}: {ISO6093 data type}

Unit: {Unit of Measure}

Example: {Attribute Field Example}

Usage Notes: {Attribute field usage notes}

{Attribute Field Name}
{Attribute Field Description}

Value {single or multiple type}: {ISO6093 data type}

Unit: {Unit of Measure}

Example: {Attribute Field Example}

Usage Notes: {Attribute field usage notes}
:

<next {Attribute Field Name}>
:

<next {Attribute Name}>
:

2.1.2 Syntax Definition

This section provides the key to bracketed ({...}) information shown in Section 2.1.1:

VCT 1 Section Name
Name of the descriptor category corresponding to VCT 1 section names.

Descriptor
A VC characteristic described by one or more VC attributes.

Section Number
The section number in VCT 1 [See Section 1.7; Reference 2 of this document] where the descriptor was first defined.

Descriptor explanation
Outlines the type of information to be included within the named descriptor category.

Attribute Name
: Name of the VC feature or characteristic being defined.

Attribute Description
: Sets forth the type of information to be documented using the attribute and indicates if attribute is a compound type.

Attribute Field Name
: Name of field being documented for the attribute. When an attribute has more than one field, all fields must be used together for a complete description of the attribute.

Attribute Field Description
: Provides a terse semantic definition of the particular field that must be limited to a single value. In the case of fields that are multiple type, the Attribute Field Name is repeated each time an additional value is associated with it.

Single or Multiple Type
: Indicates whether the field is single type and used only once (per VC description or compound attribute field, if an element of a compound attribute) or multiple type and used more than once (per VC description or compound attribute field, if an element of a compound attribute).

Unit of Measure
: Unit of measure to be used in specifying the value of the attribute field (if applicable).

ISO6093 Data Type
: Defines the syntax allowed for the field using the ISO 6093 format. (See Appendix E for details of this ISO Standard.)

Attribute Field Example
: Supplied to improve understanding of type of information typically described by the field.

Attribute Field Usage Notes
: Indicate additional information helpful for properly qualifying the data presented in the attribute field.

2.1.3 Attribute Characteristics

Attributes possess one or more data fields, called attribute fields, and are represented as colon-separated pairs when referred to as a unit. (For example, in the attribute “Package:VCID,” “Package” is the attribute name and “VCID” is the attribute field name.) Attributes themselves may be simple or compound. In simple attributes, the attribute fields take on values independently. In compound attributes, the attribute fields form an ordered array in which each individual attribute field maintains its association with its related attribute fields so that application of the attribute to successive data sets preserves the original relationships. A compound attribute may be thought of as a vector in which the attribute fields are an ordered set of values that go together. “FlipFlopCount” is an example of a compound attribute because it has a “Value” attribute field for each clock domain of the VC and must be correctly paired with its associated “ClockDomain” attribute field in order to accurately estimate VC power consumption. If an attribute is a compound type, this is noted in the *Attribute Description* text. (See Section 2.1.1, “Attribute Structure.”)

Individual attribute fields of simple attributes may be limited to a single value per VC or have multiple values assigned to them. Each attribute field indicates whether it is “single” or “multiple” in the *Value* designator. (See Section 2.1.1, “Attribute Structure.”) “Package:VCID” and “Class:Description” are examples of single-attribute fields. Multiple valued attributes fields (such as “Clocks:Networks”) take on more than one value per VC. In this case, the attribute field is paired with the first value, then paired with the next value, and so forth until each network has been described. Note that attribute fields of compound attributes can only be singly valued in each ordered group of values.

The set of descriptors and attributes documented herein form a common set of terminology and nomenclature that should be used wherever VCs are described for search, selection, and transfer purposes.

2.1.4 Remark Option

The Remark attribute field is an optional attribute field associated with one of the attributes defined in this standard. It enables the VC provider to include VC-specific information that does not fit within the existing complement of attribute fields. This document calls for remarks in specific cases, such as in the Date attribute field of the Availability attribute, which indicates information about future release dates by version in a Remark attribute field. A Remark attribute field should also be used any time the value "other" is chosen from a pick list to explain the relevant details pertinent to the particular VC being described. It is left to the VC provider's discretion to insert Remark attribute fields within other attributes as needed to document VC-specific details deemed to be essential for a complete description of the VC. The Remark attribute field has the following format:

Remark:
Free text field which may be associated with any attribute field to provide supplementary information needed to complete the attribute information set.
Value single: M..256
Example: Version 2.2 20001225, Version 3.0 20010630
Usage Notes: Optional attribute field that may be associated with any attribute that requires it individually.

2.2 Attribute Descriptions and Value Definitions

The attributes described in this section are summarized in Table 1of this document. Where a Pick list is indicated in the Usage Notes of an attribute field, consult Appendix C for values contained in the pick list. Appendix D summarizes the attribute field details in a fashion designed to aid implementers.

2.2.1 VC Provider Claims: Functional Overview

VCT 1 Section 2.1.1

The functional overview summarizes the key functions of the VC in a concise, compact style. This enables potential VC users in the search, evaluate, and select mode to rapidly determine the applicability of the VC being described.

2.2.1.1 Class

The Class attribute illustrates the general functional behavior category and classification of the VC.

Description:
Text description of the functional behavior of the VC.
Value single: M..256
Unit:
Example: interpretive JAVA execution engine with enhanced class library.
Usage Notes:

Classification:
VC category or categories within the VCT functional classification scheme.
Value multiple: M..128
Unit:
Example: processors MPU RISC 64 bit
Usage Notes: pick list (See Appendix A.)

2.2.1.2 Architecture

The Architecture attribute describes the general implementation scheme of the VC.

Description:
Text description of the VC functional implementation scheme.
Value single: M..256
Unit:
Example: two-issue superscalar construct with register renaming
Usage Notes: Use of architecture diagram is encouraged to supplement the written description.

2.2.1.3 ModesOfOperation

This attribute describes the modes of operation that cause the VC to change its functional behavior.

Description:
Explanation of the provided modes and the functional behavior of the VC in each mode.
Value multiple: M..256
Unit:
Example: Communications modes include serial, hdlc, sdlc, synchronous, asynchronous, test, debug.
Usage Notes:

2.2.1.4 EquivalentPartsOrCores

VCs are often implemented to duplicate or incorporate functionality of existing parts or VCs. This attribute provides information about these equivalent parts and cores.

Description:
List of the existing physical components and VCs whose functional behavior is equivalent to the VC.
Value multiple: M..256
Unit:
Example: code - and pin-compatible with Company A's 12345.
Usage Notes:

2.2.2 VC Provider Claims: Target Applications

VCT 1 Section 2.1.2

The VC provider shall include details of the types of applications targeted by the VC and provide actual design examples, as appropriate, to aid the VC users in their evaluation and selection process.

2.2.2.1 MarketSegments

This attribute provides information on applicable VC market segment(s), facilitating better searchability based on end-user application(s).

Description:
Description of the intended application(s) or applicable field(s) of VC end products.
Value single: M..256
Unit:
Example: consumer video
Usage Notes:

Classification:
Most closely aligned application based on the VCT market segment classification scheme
Value multiple: M..128
Unit:
Example: Consumer electronics DVD video and audio
Usage Notes: pick list (see Appendix B)

2.2.3 VC Provider Claims: Performance

VCT 1 Section 2.1.3

Performance claims enable the VC user to assess how well a particular VC can meet the overall performance goals of the VC user's design. The performance descriptor includes many different types of attributes depending on the functionality and characteristics of the VC. Those included below are representative of typical performance parameters found in many VCs.

2.2.3.1 Frequency

This attribute indicates the clock speed or data rate at which the VC operates. The value of this attribute should represent the primary operation speed of the VC.

Minimum:
The minimum value of the primary operation speed of the VC.
Value single: NR3..1.12ES..2
Unit: Hertz
Example: 6.2207E+08
Usage Notes:

Typical:
The typical value of the primary operation speed of the VC.
Value single: NR3..1.12ES..2
Unit: Hertz
Example: 6.2208E+08
Usage Notes:

Maximum:
The maximum value of the primary operation speed of the VC.
Value single: NR3..1.12ES..2
Unit: Hertz
Example: 6.2209E+08
Usage Notes:

2.2.3.2 PowerConsumption

This attribute indicates the power consumption of the VC at specified operation frequency. Minimum, typical, and maximum fields are provided.

Minimum:
The minimum value of the power consumption of the VC.
Value single: NR3..1.4ES..2
Unit: Watt
Example: 4.2E+00
Usage Notes: Specify operating voltage, frequency, and any other relevant operating conditions in a remark.

Typical:
The typical value of the power consumption of the VC
Value single: NR3..1.4ES..2
Unit: Watt
Example: 5.0E+00
Usage Notes: Specify operating voltage and frequency in a remark.

Maximum:
The maximum value of the power consumption of the VC.
Value single: NR3..1.4ES..2
Unit: Watt
Example: 7.3E+00
Usage Notes: Specify operating voltage and frequency in a remark.

2.2.3.3 Throughput

Throughput is an important measure of VC performance and represents the rate at which the VC produces or processes data. Throughput should be from actual measured results, if available, but may be a claimed result if measured data is not available. (Note which is the case in a remark.)

bps:
Throughput represented in bits per second.
Value single: NR3..1.3ES..2
Unit: bits per second
Example: 1.0E+09
Usage Notes:

fps:
Throughput represented in frames per second.
Value single: NR3..1.3ES..2

Unit: frames per second
Example: 5.0E+07
Usage Notes:

MIPS:
Throughput represented in millions of instructions per second.
Value single: NR3..1.3ES..2
Unit: million instructions per second
Example: 2.3E+03
Usage Notes:

sps:
Throughput rate represented in samples per second.
Value single: NR3..1.3ES..2
Unit: samples per second
Example: 2.5E+10
Usage Notes:

2.2.3.4 Latency

This attribute describes the maximum number of input data signals consumed before the first output is produced. It must take into account all data inputs and be described in terms of the frequency of the block or the absolute time measure wherever possible. This attribute applies to such VCs as memory and data converter. This is a compound-type attribute, which expresses latency by operating mode and interface.

Mode:
The mode of VC operation for which the Latency is applicable.
Value single: M..64
Unit:
Example: boot-up sequence.
Usage Notes:

SystemLevelInterface:
Name the specific system-level interface associated with the mode.
Value single: M..64
Unit:
Example: Data_IO.
Usage Notes:

clock:
Latency from input to output measured in terms of system clock cycles.
Value single: NR3..1.3ES..2
Unit: clock cycle
Example: 2.0E+00
Usage Notes:

sec:
Latency from input to output measured in seconds.
Value single: NR3..1.3ES..2
Unit: second
Example: 5.2E-09
Usage Notes:

2.2.4 VC Provider Claims: Form Information

VCT 1 Section 2.1.4

This section describes the hardness of the VC (soft, firm, or hard). Include the estimated or exact gate count, pin counts (input, output, and test), and area dimensions wherever possible.

2.2.4.1 Hardness

This is the fundamental VC attribute that differentiates the integration and adoption process of one VC from another. The VC provider should clearly state the hardness of the VC, as all VSIA specifications are based on this criteria.

Classification:
Classification of the VC according to the VSIA standard list of hardness levels.
Value single: A..4
Unit:
Example: soft
Usage Notes: pick list
(Soft | Firm | Hard | AMS)

2.2.4.2 GateCount

GateCount is defined to be the equivalent number of NAND gates for the design in the specific library used by the VC provider for verification. GateCount should be provided to enable users to estimate the VC die area and cost and to assess the impact of the VC on the overall chip design.

Minimum:
The minimum equivalent number of NAND gates.
Value single: NR1..10
Unit: gate
Example: 750000
Usage Notes:

Typical:
The typical equivalent number of NAND gates.
Value single: NR1..10
Unit: gate
Example: 800000
Usage Notes:

Maximum:
The maximum equivalent number of NAND gates.
Value single: NR1..10
Unit: gate
Example: 850000
Usage Notes:

2.2.4.3 BitCount

BitCount is defined to be the number of memory bits in the VC. BitCount should be provided to enable users to estimate the VC die area and cost, and to assess the impact of the VC on the overall chip design.

Minimum:
The minimum number of bits.
Value single: NR1..10
Unit: bit
Example: 8388608
Usage Notes:

Typical:
The typical number of bits
Value single: NR1..10
Unit: bit
Example: 12582912
Usage Notes:

Maximum:
The maximum number of bits.
Value single: NR1..10

Unit: bit
Example: 16777216
Usage Notes:

2.2.4.4 TotalArea

This attribute should be provided for hard and AMS VCs. For soft or firm VCs, indicate in a remark if estimated, or provide the area resulting from a verification run and/or previous implementation run(s).

Minimum:
The minimum implemented/estimated area for the entire VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 2.25E+01
Usage Notes:

Typical:
The typical implemented/estimated area for the entire VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 2.5E+01
Usage Notes:

Maximum:
The maximum implemented/estimated area for the entire VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 2.75E+01
Usage Notes:

2.2.4.5 LogicArea

This attribute should be provided for hard and AMS VCs. For soft or firm VCs, indicate in a remark if estimated, or provide the area resulting from a verification run and/or previous implementation run(s).

Minimum:
The minimum implemented/estimated area of logic in the VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 2.0E+01
Usage Notes:

Typical:
The typical implemented/estimated area of logic in the VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 2.25E+01
Usage Notes:

Maximum:
The maximum implemented/estimated area of logic in the VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 2.5E+01
Usage Notes:

2.2.4.6 MemoryArea

This attribute should be provided for hard and AMS VCs. For soft or firm VCs, indicate if estimated or provide the area resulting from a verification run.

Minimum:

The minimum implemented/estimated area of memory in the VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 9.0E+01
Usage Notes:

Typical:
The typical implemented/estimated area of memory in the VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 1.0E+02
Usage Notes:

Maximum:
The maximum implemented/estimated area of memory in the VC.
Value single: NR3..1.3ES..2
Unit: mm^2
Example: 1.1E+02
Usage Notes:

2.2.4.7 PinCount

This attribute provides the number of I/Os (pins) of various types that access the VC.

Input:
The number of input pins.
Value single: NR1..5
Unit:
Example: 46
Usage Notes:

Output:
The number of output pins.
Value single: NR1..5
Unit:
Example: 3
Usage Notes:

Bidirectional:
The number of bi-directional pins.
Value single: NR1..5
Unit:
Example: 32
Usage Notes:

Test:
The number of test pins.
Value single: NR1..5
Unit:
Example: 6
Usage Notes:

Other:
The number of other types of pins. Indicate the types in a remark.
Value single: NR1..5
Unit:
Example: 0
Usage Notes: Explain other types of pins in a remark.

2.2.4.8 FlipFlopCount

This attribute contains the number of flip-flops per clock domain, and is useful for estimating power consumption for soft and/or firm VCs. This is a compound-type attribute, which expresses flip-flop counts by clock domain name.

Value:
The number of flip-flops in the named clock domain.
Value single: NR1..10
Unit:
Example: 200000
Usage Notes:

ClockDomain:
The name of the VC clock domain for the reported flip-flop count.
Value single: M..64
Unit:
Example: A_900
Usage Notes:

2.2.5 VC Provider Claims: Test Coverage

VCT 1 Section 2.1.5

This descriptor provides information on the quality of the verification test bench and manufacturing test bench in terms appropriate to the technology of the VC.

2.2.5.1 StuckFaultCoverage

This attribute is the stuck fault coverage of the test vectors supplied with the VC. Coverage data should be provided for each manufacturing test method utilized.

Name:
The name of the particular production test suite being referenced.
Value multiple: M..64
Unit:
Example:
Usage Notes:

Value:
The numeric value of the percent total stuck fault coverage obtainable assuming use of all test suites.
Value single: NR2..3.3
Unit: percent
Example: 98.330
Usage Notes:

2.2.5.2 CodeCoverage

Code coverage should be specified in percentage terms as applicable.

Branch:
Branch and condition coverage.
Value single: NR2..3.3
Unit: percent
Example: 97.54
Usage Notes:

Variable:
Variable trace and variable toggle coverage.
Value single: NR2..3.3
Unit: percent
Example: 100
Usage Notes:

Statement:
Statement Coverage (RTL source statement coverage with the test bench provided with the VC).
Value single: NR2..3.3
Unit: percent
Example: 100
Usage Notes:

FSM_arc:
The Arc coverage obtainable for internal finite state machines (FSM).
Value single: NR2..3.3
Unit: percent
Example: 96.2
Usage Notes:

FSM_expression:
The Expression coverage obtainable for internal finite state machines (FSM).
Value single: NR2..3.3
Unit: percent
Example: 95
Usage Notes:

FSM_state:
The Visited state coverage for internal finite state machines (FSM).
Value single: NR2..3.3
Unit: percent
Example: 90
Usage Notes:

2.2.6 VC Provider Claims: List of Deliverables

VCT 1 Section 2.1.6

This section provides a comprehensive itemization of all the component parts of the VC.

2.2.6.1 Deliverables

This attribute lists all deliverables included in the VC provider package using the table from the current version of *VSI Alliance Deliverables Document* as a checklist. A compound-type attribute is used to describe the table format.

Name:
Concatenate VSIA document, version, section, and deliverable name as listed in Table 1 of the current version of *VSI Alliance Deliverables Document.*
Value single: M..256
Unit:
Example: I/V 1 2.0 Section 2.1.1 RTL Source
Usage Notes: If the deliverable is not specified by VSIA, choose an appropriate name different from existing VSIA deliverable names and provide an explanation in the remark.

Comply:
Indicates whether the named deliverable is present in a form compliant with the applicable VSIA standard (if defined for this type of VC).
Value single: A..12
Unit:
Example: Y
Usage Notes: pick list
(Y N Does_Not_Apply)

Format:
The data format, applicable standard, and version of the format.
Value single: M..64
Unit:

Example: VHDL IEEE1076-1987
Usage Notes: Concatenate format and format version.

Filename:
The name of the file containing the named deliverable.
Value single: M..64
Unit:
Example: F5003V.vhdl
Usage Notes:

Description:
A description of the deliverable.
Value single: M..256
Unit:
Example: RTL source for the codec portion of VC
Usage Notes:

2.2.6.2 Availability

This attribute indicates that the VC is currently available for transfer to VC users or provides a target date for availability. A VC is available when the specified deliverables in the Deliverable List attribute can be distributed to users.

Date:
The date (in YYYYMMDD format) on which the declared deliverables became or will become available to users.
Value single: N 8
Unit: yyyymmdd
Example: 20001023
Usage Notes: Put future release dates by version in remark.

2.2.6.3 Reference

Identifies the *VSI Alliance Deliverables Document* version applicable to the deliverables described herein.

Version:
The date (in YYYYMMDD format) on which the declared deliverables became or will become available to users.
Value single: N 8
Unit: yyyymmdd
Example: 20001023
Usage Notes:

2.2.7 VC Provider Claims: Features and Standards Compliance

VCT 1 Section 2.1.7

This section lists the main features and benefits of the VC, states the standards with which the VC is compliant, and indicates the specific revision of each standard applicable to the VC.

2.2.7.1 CompliantStandard

This attribute contains the name(s) of all standard(s) with which the VC is intended to comply. This is a compound-type attribute, providing support for as many standards as are applicable to the VC being described.

Organization:
The name of the organization that published the standard.
Value single: M..64
Unit:
Example: IEEE
Usage Notes:

Name:
The identifier and title of the standard.

Value single: M..64
Unit:
Example: IEEE1394
Usage Notes:

Version:
The release version number of the standard applicable to the VC being described.
Value single: M..32
Unit:
Example: 1.2
Usage Notes:

Revision:
Where used, the release revision number indicating the specific editorial revision level of the standard.
Value single: M..32
Unit:
Example: a 4
Usage Notes:

2.2.7.2 Features

This attribute provides a means for VC providers to list additional information (such as key words or detail that does not fit under any other attribute) to provide additional differentiation of the VC.

Description:
A list of key words characterizing the VC.
Value multiple: M..256
Unit:
Example: Supports multi-master systems with automatic arbitration and clock synchronization, provision interrupt, and address detection.
Usage Notes:

2.2.8 System/Logic Description: Abstract Models

VCT 1 Section 2.2.2

This section contains information about models, their abstraction level, how they are used in the design flow, and their accuracy (cycle vs. instruction). For more detailed information, consult (Section 1.7; Reference 4), *VSI Alliance System-Level Interface Behavioral Documentation Standard*.

2.2.8.1 ModelingAbstractions

Types of abstract models provided with the VC, listed from most abstract to most detailed. Internal resolution references how a model describes timing of events, function, values, and structures of the elements that are contained within the boundaries of the modeled device.

ModelName:
Name of Model
Value multiple: M..64
Unit:
Example: Viterbi Bus-Functional Model
Usage Notes:

ModelType:
Description of model types using the categories set forth in *VSI Alliance System-Level Design Model Taxonomy version 1.0.*
Value single: M..32
Unit:
Example: Token_Cycle_Accurate
Usage Notes: pick list
(Partially_Ordered_Event_Accurate | System_Event_Accurate | Token_Cycle_Accurate | Instruction_Cycle_Accurate | Cycle_Approximate_Accurate | Cycle_Accurate | Gate_Propagation_Accurate)

ModelResolutionInternalTemporal:
Represents the degree of accuracy of events modeled along a time scale or in time that are contained within the boundaries of the modeled device.
Value single: M..17
Unit:
Example: Instruction_Cycle
Usage Notes: pick list
(Gate_Propagation | Clock_Accurate | Cycle_Approximate | Instruction_Cycle | Token_Cycle | System_Event | Partial_Order)

ModelResolutionInternalData:
Represents the level of detail with which a model specifies the format of values that are contained within the boundaries of the modeled device.
Value single: A..8
Unit:
Example: Value
Usage Notes: pick list
(Bit | Format | Value | Property | Token)

ModelResolutionInternalFunctional:
Represents the level of detail with which a model describes the functionality of a component or system that is contained within the boundaries of the modeled device.
Value single: A..13
Unit:
Example: Algorithmic
Usage Notes: pick list
(Digital_Logic | Algorithmic | Mathematical)

ModelResolutionInternalStructural:
Represents the level of detail with which a model describes how a component interface is constructed.
Value single: A..13
Unit:
Example: Black_Box
Usage Notes: pick list
(Structural | Block_Diagram | Black_Box)

ModelResolutionExternalTemporal:
Represents the degree of accuracy of events modeled along a time scale or in time with respect to how the model describes the interface of the modeled device to other devices.
Value single: M..17
Unit:
Example: Cycle_Approximate
Usage Notes: pick list
(Gate_Propagation | Clock_Accurate | Cycle_Approximate | Instruction_Cycle | Token_Cycle | System_Event | Partial_Order)

ModelResolutionExternalData:
Represents the level of detail with which the format of values is specified in a model with respect to how the model describes the interface of the modeled device to other devices.
Value single: A..8
Unit:
Example: Bit
Usage Notes: pick list
(Bit | Format | Value | Property | Token)

ModelResolutionExternalFunctional:
Represents the level of detail with which a model describes functionality of a component or system with respect to how the model describes the interface of the modeled device to other devices.
Value single: A..13
Unit:
Example: Digital_logic

Usage Notes: pick list
(Digital_Logic | Algorithmic | Mathematical)

ModelResolutionExternalStructural:
Represents the level of detail at which the model describes the interface of the modeled device to other devices.
Value single: A..13
Unit:
Example: Structural
Usage Notes: pick list
(Structural | Block_Diagram | Black_Box)

ModelResolutionSoftwareProgramming:
Represents the granularity of software instructions that the model of a hardware component interprets in executing target software.
Value single: M.. 14
Unit:
Example: NULL
Usage Notes: pick list
(Object_Code | Micro_Code | Assembly_Code | HLL_Statements | DSP_Primitive | Major_Modes | NULL)

LanguageSupport:
Indicates the language in which the model is expressed.
Value single: M.. 64
Unit:
Example: Timed_C
Usage Notes:

ModelOfComputation:
Refers to the underlying basis for the model and the way in which modeling variables are treated. Types of models include CSP - concurrent threads with rendezvous, CT - continuous-time modeling, DE - discrete-event systems, DT - discrete time (cycle driven), PN - process networks, SDF - synchronous dataflow, and SR - synchronous/reactive .
Value single: A.. 3
Unit:
Example: DE
Usage Notes: pick list
(CSP | CT | DE | DT | PN | SDF | SR)

ToolName:
Lists tool names, versions, and formats in free text.
Value single: M.. 256
Unit:
Example: Affirma NC Verilog release 1.1
Usage Notes: Concatenate name, format, and version.

2.2.9 System/Logic Description: Structural Views

VCT 1 Section 2.2.3

The description of the architecture of the VC as seen from the outside, and information about the internal workings that are observable during simulation.

2.2.9.1 InterfaceStructure

Defines the list of interfaces for the object.

InterfaceName:
Names the particular interface being described.
Value single: M..64
Unit:
Example: System_Bus_Interface

Usage Notes:

InterfaceDescription:
Description of pertinent detail about the named interface.
Value single: M..256
Unit:
Example: Interface at implementation level is the AMBA Bus
Usage Notes:

2.2.9.2 InternalObservabilityStructure

Defines the variables that are internally observable during simulation.

InternalVarName:
Name of the variable.
Value single: M..64
Unit:
Example: Interrupt_Flags
Usage Notes:

InternalVarField:
The data format of the internal variable.
Value single: M..64
Unit:
Example: vsi_bitvector
Usage Notes: pick list
(undefined | file | record | vsi_int8 | vsi_int16 | vsi_int32 | vsi_int64 | vsi_uint8 | vsi_uint16 | vsi_uint32 | vsi_uint64 | vsi_string | vsi_bit | vsi_mvl | vsi_bitvector | vsi_mvlvector | vsi_signed | vsi_unsigned | vsi_fxval | vsi_fxnum | vsi_Tbitvector | vsi_Tmvlvector | vsi_Tsigned | vsi_Tunsigned | vsi_fixed | vsi_ufixed)

InternalVarDescription:
Describes anything pertinent to the understanding of the Internal Variable.
Value single: M..256
Unit:
Example: List of pending interrupts to service
Usage Notes:

2.2.10 System/Logic Description: System-Level Interfaces

VCT 1 Section 2.2.4.1

The system-level interface resides on the block boundary of abstract VC models. It is often constructed with layers of communication abstraction. These layers depend upon the objectives of the VC model and the stage of system-level design in which the models are utilized.

2.2.10.1 PortDescriptions

This attribute gives a system-level view of each port and describes the kinds of transactions that are valid, the way the data flows, and how each port is used.

PortName:
Name of the Port.
Value single: M..64
Unit:
Example: Data_out
Usage Notes:

InterfaceName:
Refers to the name of the system-level interface described in Section 2.2.9, “Structural Views,” for which this port is a part.
Value single: M..64
Unit:
Example: System_bus_interface

Usage Notes:

PortTypeControl:
Describes the type of control available on the port (if any) in free text.
Value single: M..9
Unit:
Example: Initiator
Usage Notes: pick list
(Null (No_Control) | Responder | Initiator)

PortTypeData:
Describes how the port handles data.
Value single: M..9
Unit:
Example: Producer
Usage Notes: pick list
(Null (No_Data) | Consumer | Producer)

TransactionType:
Indicates what actions the port performs.
Value single: M..16
Unit:
Example: transCloseChannel, transSynchronize.
Usage Notes: pick list
(transRead | transWrite | messSense | messEmit | transOpenChannel | transCloseChannel | transSynchronize | transReset | transControl | messRead | messWrite)

PortFlow:
Identifies port behavior, such as data persistence and blocking.
Value multiple: M..22
Unit:
Example: Persistent
Usage Notes: pick list
(Persistent | Buffered | FIFO | LIFO | Blocking | Assigned_Port_Priority | Assigned_Data_Priority | Multi_Rate | Pipelined | Exceptions_Handled)

Description:
Declares the purpose of the port within the named system-level interface.
Value single: M..256
Unit:
Example: Output port including address assignment for outgoing data
Usage Notes:

2.2.11 System/Logic Description: Logical (Mapped) Interfaces

VCT 1 Section 2.2.4.2

Logical (mapped) interface attributes apply to a VC interface when the VC is mapped to register transfer level (RTL) or below.

2.2.11.1 LogicInterfaceType

This attribute describes interfaces to any of the existing busses or interfaces in the industry.

Description:
List of the supported interfaces in free text.
Value single: M..256
Unit:
Example: processor abc-compatible, 64-bit bus interface with arbitration scheme
Usage Notes:

2.2.11.2 Ports

This attribute describes each interface port on the VC. It is a compound-attribute type. Multiple instances must be described.

Name:
The name of the port.
Value single: M..64
Unit:
Example: master_address
Usage Notes:

ApplicationType:
The way the port is used.
Value single: A..7
Unit:
Example: Address
Usage Notes: pick list
(Address | Data | Control | Status | Other)

DataType:
The data format handled by the port.
Value single: M..32
Unit:
Example: vsi_mvlvector
Usage Notes: pick list
(undefined | file | record | vsi_int8 | vsi_int16 | vsi_int32 | vsi_int64 | vsi_uint8 | vsi_uint16 | vsi_uint32 | vsi_uint64 | vsi_string | vsi_bit | vsi_mvl | vsi_bitvector | vsi_mvlvector | vsi_signed | vsi_unsigned | vsi_fxval | vsi_fxnum | vsi_Tbitvector | vsi_Tmvlvector | vsi_Tsigned | vsi_Tunsigned | vsi_fixed | vsi_ufixed | enum_vsi_q_mode | enum_vsi_o_mode | enum_vsi_switch_mode | enum_vsi_context_begin | enum_vsi_numrep | enum_vsi_fmt)

Direction:
The direction of the named port.
Value single: A..13
Unit:
Example: Output
Usage Notes: pick list
(Input | Output | Bidirectional)

Width:
The signal count of the port.
Value single: NR1..4
Unit:
Example: 64
Usage Notes:

Description:
A description of the port.
Value single: M..256
Unit:
Example:
Usage Notes:

2.2.11.3 ConfigurationRegisters

This attribute describes any registers required to set the VC into a specific mode of operation.

Present:
Existence of one or more configuration registers on the VC.
Value single: M..3
Unit:
Example: yes

Usage Notes:

Description:
The details of the operation of any configuration registers on the VC.
Value single: M..256
Unit:
Example: Bits 2 and 3 of register row x0a21 control the read latency of the memory output.
Usage Notes:

2.2.11.4 InterruptInterface

Interfaces used to interrupt the VC.

Name:
The name of the particular InterruptInterface.
Value single: M..64
Unit:
Example: I_Interface_3
Usage Notes:

Description:
Details the behavior and the span of control for the InterruptInterface.
Value single: M..256
Unit:
Example:
This interface interrupts communication across the ATM data ports based on congestion on the Ethernet port on the same VC.
Usage Notes:

Type:
Expresses the type of interrupt.
Value single: M..64
Unit:
Example: Round-robin.
Usage Notes:

InterruptPort:
Identifies all particular ports dedicated to the interrupt function and part of the named InterruptInterface.
Value single: M..64
Unit:
Example: Priority port.
Usage Notes:

2.2.11.5 ArbitrationInterface

Interfaces used to resolve data contention on any port(s).

Name:
The name of the particular ArbitrationInterface.
Value single: M..64
Unit:
Example: A_Interface_1
Usage Notes:

Description:
Details the behavior and the span of control for the ArbitrationInterface.
Value single: M..256
Unit:
Example: The arbitration scheme controlled by this interface applies only to data busses 1 and 2.
Usage Notes:

Type:
Expresses the class of Arbitration.
Value single: M..64
Unit:
Example:
Usage Notes:

ArbitrationPort:
Identifies all particular ports dedicated to the arbitration function and part of the named ArbitrationInterface.
Value single: M..64
Unit:
Example: Arb_1
Usage Notes:

2.2.11.6 MemoryAllocation

Details about the configuration and the flexibility of the memory available within the VC.

StartAddress:
Provides the start address of memory block
Value single: X..12
Unit: hexadecimal
Example: 004AD0
Usage Notes: If no start address is provided, memory is relocatable.

MinimumSize:
The smallest allocatable block of the available memory.
Value single: NR3..1.14E..15
Unit: byte
Example: 4.096E3
Usage Notes: The number of bytes in the smallest allocatable block.

MaximumSize:
The largest allocatable block of the available memory.
Value single: NR3..1.14E..15
Unit: byte
Example: 4.295E9
Usage Notes:

2.2.12 System/Logic Description: Integration Requirements

VCT 1 Section 2.2.5

Implementation platform requirements not adequately covered elsewhere.

2.2.12.1 APISupport:

Details about any Application Programming Interfaces available for the VC.

APIName:
The name of the API.
Value single M.. 64
Unit:
Example: API_OMI
Usage Notes:

APIStandardReference:
The title and/or document number of the published source which controls content of API.
Value single: M..256
Unit:
Example:
Usage Notes:

APIDescription:
Provides relevant application information about the API.
Value single: M..256
Unit:
Example: Standard API for HDL co-sim
Usage Notes:

2.2.13 System/Logic Description: System/Logic Test Suite

VCT 1 Section 2.2.6

This section describes the tools supplied by the VC provider to verify the correct functionality of the VC in the user design environment and should include comprehensive instruction for utilizing any verification test bench provided.

2.2.13.1 TestSuites

The external environment of VC evaluation which includes all of the interfaces and data gathering elements necessary to fully verify VC functionality

SuiteName:
Name of suite being described.
Value multiple: M..256
Unit:
Example: <name>
Usage Notes:

SuiteResolutionExternalData:
Indicates the type of data values addressed by the named test suite.
Value single: A..8
Unit:
Example: Bit
Usage Notes: pick list
(Bit | Format | Value | Property | Token)

SuiteResolutionExternalFunctional:
Describes the functional level at which the named test suite model operates.
Value single: A..13
Unit:
Example: Digital_Logic
Usage Notes: pick list
(Digital_Logic | Algorithmic | Mathematical)

SuiteResolutionExternalStructural:
Indicates the type of structural elements represented by the named test suite model.
Value single: A..13
Unit:
Example: Structural
Usage Notes: pick list
(Structural | Block_Diagram | Black_Box)

2.2.14 Physical Description: Timing Specification

VCT 2.3.1.1

The Timing Specification descriptor may include many different types of attributes depending upon the functional characteristics of the VC. The attributes included below are representative of typical physical timing specification attributes found in many VCs. If the VC provider requires additional attributes to express the timing specification of the VC, a similar style of presentation is recommended.

2.2.14.1 Duty

This attribute indicates the proportion of time that a signal is active versus inactive during each cycle. This attribute may be used for both input and output signals on the VC boundary, such as the duty cycle requirement for the input system clock.

Minimum:
The minimum expected duty cycle.
Value single: NR2..3.3
Unit: percent
Example: 45
Usage Notes:

Typical:
The typical expected duty cycle.
Value single: NR2..3.3
Unit: percent
Example: 50
Usage Notes:

Maximum:
The maximum expected duty cycle.
Value single: NR2..3.3
Unit: percent
Example: 55
Usage Notes:

2.2.14.2 Period

This attribute is the cycle time of a periodic signal such as a clock. This attribute can be used for both input and output signals on the VC boundary.

Minimum:
The minimum expected period.
Value single: NR3..1.12ES..2
Unit: second
Example: 1.1E-09
Usage Notes:

Typical:
The typical expected period.
Value single: NR3..1.12ES..2
Unit: second
Example: 1.15E-09
Usage Notes:

Maximum:
The maximum expected period.
Value single: NR3..1.12ES..2
Unit: second
Example: 1.2E-09
Usage Notes:

2.2.14.3 Phase

This attribute indicates the delay of the signal relative to the start of the relevant clock cycle. This attribute can be used for both input and output signals on the VC boundary.

Minimum:
The earliest expected signal validity.
Value single: NR1..2
Unit: percent
Example: 15

Usage Notes:

Typical:
The typical expected signal validity
Value single: NR1..2
Unit: percent
Example: 19
Usage Notes:

Maximum:
The maximum expected signal validity.
Value single: NR1..2
Unit: percent
Example: 23
Usage Notes:

2.2.15 Physical Description: Electrical Characteristics

VCT 1 Section 2.3.1.2

The electrical specification of a VC should include a list of VC parameters with the definition and calculation method for each. The electrical characteristics descriptor may include many different types of attributes depending upon the functional characteristics of the VC. The attributes included under this section are representative of typical electrical characteristics attributes found in many VCs.

2.2.15.1 SupplyVoltage

This attribute describes the operating voltage necessary for the VC to operate as intended and is applicable to hard cores (and, in some cases, firm cores). This is a compound-type attribute.

Name:
The name of the supply voltage.
Value single: M..64
Unit:
Example: Vcc
Usage Notes:

Minimum:
The minimum supply voltage.
Value single: NR2..3.3
Unit: volt
Example: 1.8
Usage Notes:

Typical:
The typical supply voltage.
Value single: NR2..3.3
Unit: volt
Example: 2
Usage Notes:

Maximum:
The maximum supply voltage.
Value single: NR2..3.3
Unit: volt
Example: 2.2
Usage Notes:

2.2.15.2 OperatingTemperature

This attribute describes the ambient operating temperature range necessary for the VC to operate as intended. It is applicable to hard cores (and, in some cases, firm cores).

Minimum:
The minimum ambient operating temperature.
Value single: NR1 S..3
Unit: degree Celsius
Example: -40
Usage Notes:

Typical:
The typical ambient operating temperature.
Value single: NR1 S..3
Unit: degree Celsius
Example: 25
Usage Notes:

Maximum:
The maximum ambient operating temperature.
Value single: NR1 S..3
Unit: degree Celsius
Example: +85
Usage Notes:

2.2.15.3 SNR

This compound-type attribute describes VC signal-to-noise ratio requirements (essential for fully describing components such as data-acquisition mixed-signal VCs). It is applicable to analog and mixed signal VCs only.

SignalName:
Name of the signal whose signal-to-noise ratio is specified.
Value single: M..64
Unit:
Example: Signal1
Usage Notes:

Minimum:
The minimum expected SNR.
Value single: NR2 S..3.3
Unit: dB
Example: +6
Usage Notes: Declare measurement conditions in remark.

Typical:
The typical expected SNR.
Value single: NR2 S..3.3
Unit: dB
Example: +12
Usage Notes: Declare measurement conditions in remark.

Maximum:
The maximum expected SNR.
Value single: NR2 S..3.3
Unit: dB
Example: +30
Usage Notes: Declare measurement conditions in remark.

2.2.15.4 Sensitivity

This compound-type attribute indicates the degree of tolerance of the VC to interference such as noise.

Signalname:
Name of the signal whose sensitivity is specified.
Value single: M..64
Unit:
Example: Input1

Usage Notes:

Minimum:
The minimum expected sensitivity.
Value single: NR2 S..3.3
Unit: dB
Example: +1
Usage Notes: Declare measurement conditions in remark.

Typical:
The typical expected sensitivity.
Value single: NR2 S..3.3
Unit: dB
Example: +2
Usage Notes: Declare measurement conditions in remark.

Maximum:
The maximum expected sensitivity.
Value single: NR2 S..3.3
Unit: dB
Example: +3
Usage Notes: Declare measurement conditions in remark.

2.2.15.5 I/O_Characteristics

This compound attribute describes the Input/Output signals on the VC boundary.

Name:
Name of the I/O being described.
Value single: M..64
Unit:
Example:
Usage Notes:

Type:
The type of I/O being described.
Value single: M..16
Unit:
Example: DTL, TTL, PCI, LVDS, Differential, Single Ended, ECL, PECL
Usage Notes:

2.2.15.6 OutputCharacteristic

This compound attribute describes the measurement of output signals on the VC boundary.

Signalname:
Name of the output being measured.
Value single: M..64
Unit:
Example: RY/BY#
Usage Notes:

Metric:
The name of the measured quantity.
Value single: M..32
Unit:
Example: tBUSYmin
Usage Notes:

Value:
The numeric value of the measurement.
Value single: NR3..3.3ES..2
Unit:

Example: 9.0E-08
Usage Notes: Declare measurement conditions in remark.

UnitOfMeasure:
The unit of measure.
Value single: M..32
Unit:
Example: seconds
Usage Notes:

2.2.16 Integration Requirements: Clock Distribution

VCT 1 Section 2.3.2.1

This descriptor provides information about clocks and clock distribution including methods of and conditions for specification.

2.2.16.1 Clocks

This attribute describes the type of clock network implemented or suggested for this VC.

Networks:
Indicates a type of clock network according to a standard list. Use a remark to detail “other.”
Value multiple: A..13
Unit:
Example: Grid
Usage Notes: pick list
(Balanced_Tree | H_Tree | Grid | Other)

SkewRiseMax:
The maximum value of skew for the rising edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 1.0E-08
Usage Notes:

SkewRiseTyp:
The typical value of skew for the rising edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 1.0E-08
Usage Notes:

SkewRiseMin:
The minimum value of skew for the rising edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 1.0E-08
Usage Notes:

SkewFallMax:
The maximum value of skew for the falling edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 9.0E-09
Usage Notes:

SkewFallTyp:
The typical value of skew for the falling edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 9.0E-09
Usage Notes:

SkewFallMin:
The minimum value of skew for the falling edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 9.0E-09
Usage Notes:

SlewRise:
The typical value of slew for the rising edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 5.0E-09
Usage Notes: Expressed as transition time 25% to 75% of voltage swing.

SlewFall:
The typical value of slew for the falling edge.
Value single: NR3..1.5ES..2
Unit: second
Example: 3.0E-09
Usage Notes: Expressed as transition time 25% to 75% of voltage swing.

JitterMax:
The maximum numeric value of clock jitter.
Value single: NR2..3.3
Unit: percent of clock period
Example: 5
Usage Notes:

JitterTyp:
The typical numeric value of clock jitter.
Value single: NR2..3.3
Unit: percent of clock period
Example: 5
Usage Notes:

JitterMin:
The minimum numeric value of clock jitter.
Value single: NR2..3.3
Unit: percent of clock period
Example: 5
Usage Notes:

ActiveEdges:
Lists the edge or level according to a standard list.
Value multiple: A..12
Unit:
Example: Rising Edge
Usage Notes: pick list
(Rising_Edge | Falling_Edge | Both_Edges | High_Level | Low_Level)

2.2.17 Integration Requirements: Design Constraints

VCT 1 Section 2.3.2.2

Design constraints are specified for the VC as a whole, for hierarchical blocks within the VC, or for particular signals within the VC.

2.2.17.1 Abstract

This attribute describes the VC design constraints on the aspect ratio, and the ability to flip, mirror, or rotate the abstract in the final layout.

Ratio_Min:
The minimum aspect ratio that can be tolerated for the claims to be valid.
Value single: NR2..2.2
Unit:
Example: 2
Usage Notes: Include minimum and maximum levels separately.

Ratio_Max:
The maximum aspect ratio that can be tolerated for the claims to be valid.
Value single: NR2..2.2
Unit:
Example: 4
Usage Notes: Include minimum and maximum levels separately.

Operations:
Lists any physical operation that cannot be tolerated by the VC.
Value multiple: A..6
Unit:
Example: mirror
Usage Notes: pick list
(Rotate | Flip | Mirror)

Ratios_supported:
A list of supported aspect ratios for a hard macro.
Value multiple: NR2..2.2
Unit:
Example: 1.78
Usage Notes:

2.2.17.2 Reset

This attribute describes the global reset scheme of the VC.

Types:
Lists the type of reset as asynchronous or synchronous.
Value multiple: A..12
Unit:
Example: synchronous
Usage Notes: pick list
(Synchronous | Asynchronous | Both | None)

2.2.18 Integration Requirements: Process Requirements

VCT 1 Section 2.3.2.5

2.2.18.1 SpecifiedProcesses

This attribute provides a description of each process technology necessary to realize all claims.

Description:
Specified process description in free text.
Value multiple: M..256
Unit:
Example: 0.2u Micron, 4 levels of Al Metal
Usage Notes:

2.2.18.2 SpecifiedFoundries

This attribute includes the verified foundry and foundries that are capable of meeting VC process requirements.

Description:
List of the specified foundry names.
Value multiple: M..256
Unit:
Example: Company Z
Usage Notes:

2.2.19 Integration Requirements: Design Process Sensitivities

VCT 1 Section 2.3.2.6

2.2.19.1 ProcessSensitivity

This compound attribute documents the process parameters that affect the functionality, performance, or other characteristics of the VC.

SignalNames:
Name of signals affected by variations in process parameters.
Value single: M..64
Unit:
Example: About
Usage Notes:

Description:
Specified process description in free text.
Value single: M..256
Unit:
Example: The frequency of VC is inversely proportional to the oxide thickness in the oscillator section.
Usage Notes:

2.2.20 Integration Requirements: Implementation Test Suite

VCT 1 Section 2.3.3

This section contains information pertaining to the proof of satisfactory VC integration and physical implementation. Verification information is primarily described here.

2.2.20.1 ImplementationTestSuite

This compound-type attribute describes the implementation test suite features.

TestSuiteName:
The name of the particular test suite being described.
Value single: M..256
Unit:
Example:
Usage Notes:

Testing:
The amount of time required to run the specified validation test vectors on the specified tester.
Value single: NR2..8.2
Unit: second
Example: 6.74
Usage Notes:

TestFrequency:
Tester frequency assumed in the test time specification.
Value single: NR3..1.12ES..2
Unit: Hertz
Example: 5.6000000000000E07
Usage Notes:

VectorSize:
Contains the maximum vector size used by the test patterns to test the VC on the specified tester.

Value single: NR1..16
Unit:
Example: 67434
Usage Notes:

TesterType:
States the tester type.
Value single: M..64
Unit:
Example: Vseries 60000
Usage Notes: One tester type per suite entry; indicate any other specific requirements for the test suite in a remark.

2.2.21 Reference Environment: Verification of Claims

VCT 1 Section 2.4.1

The reference environment is the context in which all VC features, characteristics, and performance claims are established and validated. VC provider claims and assumptions are verified by actual measurement for hard VCs. For soft and firm VCs, verification is achieved by running the design through to completion against the defined reference environment.

2.2.21.1 QualificationLevels

This attribute indicates the level at which the VC has been implemented. It verifies and provides useful information to implementers, since VCs are not delivered as finished and verified physical products.

Classifications:
Classifies qualification according to a standard list.
Value multiple: A..23
Unit:
Example: Production
Usage Notes: pick list (See “Note 2” in Appendix C for detailed explanation of list items.)
(Architecture_Simulation | RTL_Verification | Gate_Verification | Formal_Verification | Static_Timing_Analysis | FPGA | Prototype | Production | Others)

2.2.21.2 Certifications

This attribute is used for indicating VC certification details. Typically, certification is described by naming the certifying organization, the certification process, and any specific standards to which the VC has been certified.

Descriptions:
Lists certification details in free text.
Value multiple: M..256
Unit:
Example: Conforms to CAN 2.0B Specification as certified by VHDL CAN Test Suite V1.2. Conforms to CAN 2.0B Specification as certified by Test House.
Usage Notes:

2.2.22 Reference Environment: Tools, Flows, and Methodology

VCT 1 Section 2.4.2

Tools, Flows, and Methodology documents the reference environment by listing all the tools used and the version and revision information for each, so that the reference environment may be exactly reproduced to whatever extent VC users desire.

2.2.22.1 EDAtools

This compound-type attribute lists each of the EDA tools in the chain used by the VC provider. It details every product and software release version number necessary to duplicate the EDA environment.

Name:
Name of the EDA tool.

Value single: M..64
Unit:
Example: Modelsim (Verilog & VHDL)
Usage Notes: Explain use of each particular tool in the chain by associating a remark with the entry.

Version:
Version of the named EDA tool.
Value single: M..32
Unit:
Example: 1999-05
Usage Notes:

2.2.22.2 SupportTools

This attribute contains details (including all version information) for any additional software tools, such as C-compilers and Module Linkers, that are required to use components like processor VCs.

Descriptions:
A list of the support tools provided with brief descriptions for each.
Value multiple: M..256
Unit:
Example: GNU C compiler version 1.07 Beta Copyright (C) 1993 Free Software Foundation, Inc.
Usage Notes:

2.2.22.3 ReferenceDesigns

This attribute contains a listing of reference designs and reference platforms such as evaluation boards, design kits, evaluation models, and evaluation chips.

Descriptions:
List of reference designs.
Value multiple: M..256
Unit:
Example: A plug-in replacement for a zz3300 is available using the AAA 0.18-micron CMOS technology; an evaluation board with full ICE is available for this design.
Usage Notes: Concatenate platform type and availability.

2.2.23 Reference Environment: ASIC Libraries

VCT 1 Section 2.4.3

This section provides a description of all the cell-library needs of the VC. It identifies the logic functions required and the critical timing, power, and drive characteristics associated with each.

2.2.23.1 Library

This compound-type attribute enumerates the library name and the vendor names.

Name:
Contains the names and version numbers of ASIC cell libraries used to verify the VC.
Value single: M..64
Unit:
Example: BBB 0.18-micron CMOS Library v1.00.
Usage Notes:

Vendor:
Contains the supplier name for the ASIC cell libraries used to verify the VC.
Value single: M..64
Unit:
Example: Company ABC
Usage Notes:

2.2.24 Reference Environment: Process Technology

VCT 1 Section 2.4.4

This section provides a description of the process technology in which the VC has been verified.

2.2.24.1 ProvenProcesses

Names the process technology in which the VC has been verified.

Descriptions:
A free text description of the key elements of the process, such as geometry, number of metal layers, metal material, and metal migration limit.
Value multiple: M..256
Unit:
Example: 0.25-micron 5LM Cu CMOS 3V Working Max Clock Speed 50 MHz Typical 0C -70C
Usage Notes:

2.2.24.2 ProvenFoundries

Descriptions:
Name(s) of any specific foundry or foundries used for verification.
Value multiple: M..64
Unit:
Example: Company X
Usage Notes:

2.2.25 Application Information: Version History

VCT 1 Section 2.5.1

VC version history should provide sufficient information to detail all pertinent changes throughout the life of the VC and clearly identify the specific release level that correlates with VC Provider claims.

2.2.25.1 Version

This attribute describes the version of the VC.

Number:
Version code specific to VC being described.
Value single: M..32
Unit:
Example: V2.70A
Usage Notes: Must be a unique identifier of VC release. Should separate major version from minor revision, order them version (left) then revision (right), and make the combined version-revision string change monotonically throughout version history for ease of sorting. For example, 002 004 (note use of leading zeroes to prevent problems when more than nine releases occur).

Date:
Date value
Value single: N 8
Unit: yyyymmdd
Example: 20000301
Usage Notes:

2.2.26 Application Information: Known Bugs

VCT 1 Section 2.5.2

Known bugs provides a record of bugs, issues, and other problems known to exist with the VC being described, including workarounds and plans for fixes (if available).

2.2.26.1 BugList

This attribute contains a succinct list of all bugs affecting the functioning of the VC.

Descriptions:
A free text listing of bugs.
Value multiple: M..256
Unit:
Example: Has not passed the number 12 and number 14 vectors of the ITU-T G.726 test suite (burst DMA transfer in mode z not functioning).
Usage Notes: The purpose of a bug list in the profiling attributes is to declare major exceptions from expected function or performance. While the fully detailed bug list is a deliverable document, bug id/bug discussion pairs can be placed here. Concatenating several bugs per entry is allowed.

2.2.27 Application Information: Application Notes

VCT 1 Section 2.5.3

Application notes are used to facilitate the VC user's adoption of the VC and to explain design implementation options.

2.2.27.1 RelatedVCs

This attribute lists related VCs within the same family along with relevant descriptive details to enable a VC user to choose the best candidate for implementing the particular application under consideration.

Descriptions:
A free text listing of VC related to the VC being described.
Value multiple: M..256
Unit:
Example: m12c345 UART, m12233 UART with TX and RX FIFOS, m99988 enhanced UART with 256 byte FIFOS and automatic handshake.
Usage Notes:

2.2.27.2 CustomizationOptions

This attribute indicates available VC options offering customization or parameterization.

Descriptions:
A free text description of customization options.
Value multiple: M..256
Unit:
Example: configurable for 32- or 64-bit data bus interfaces
Usage Notes:

2.2.28 Test Information: Test Strategy

VCT 1 Section 2.6.1

The test strategy provides a high-level overview of the recommended approach for achieving test coverage while minimizing cost factors.

2.2.28.1 TestMethods

This attribute lists the test methods used for manufacturing. For soft or firm VCs, use this attribute to describe the recommended test method(s).

Classifications:
Classifies methods from a standard list. If other, explain in a remark.
Value multiple: A..13
Unit:
Example: BIST
Usage Notes: pick list
(Partial_Scan | Full_Scan | Boundary_Scan | BIST | IDDQ | Other)

2.2.29 Test Information: Test Module

VCT 1 Section 2.6.2

A test module is an encapsulation of a test protocol that specifies precisely how a test is to be performed.

2.2.29.1 2.2.29.1 TestModule

This compound-type attribute summarizes the test modules provided for production test purposes.

Name:
The name of the particular production test suite being referenced.
Value single: M..64
Unit:
Example:
Usage Notes:

Format:
Contains the format of the test vectors used.
Value single: M..64
Unit:
Example: STIL rev x.x
Usage Notes:

Time:
Provides the amount of time required to run the specified manufacturing test vectors on the specified tester.
Value single: NR2..8.2
Unit: second
Example: 25
Usage Notes:

Frequency:
Gives the tester frequency assumed in the test time specification.
Value single: NR3..1.12ES..2
Unit: Hertz
Example: 6.0E10
Usage Notes:

Size:
Contains the maximum vector size used in the VC test patterns on the specified tester.
Value single: NR1..16
Unit:
Example: 23222
Usage Notes:

Type:
Describes the tester type used with the supplied test vectors.
Value single: M..64
Unit:
Example: company xxx, model yy
Usage Notes: One tester type per entry; indicate in a remark any unusual features that could affect the mapping or conversion of the test patterns for use on another tester type.

2.2.30 Supplier Information: VC Provider Contact Information

VCT 1 Section 2.7.1

This section contains the VC provider's complete contact details, and directions for finding additional information.

2.2.30.1 ContactInformation

This attribute provides the set of contact information of the VC provider.

Name:
Contains the name of the VC provider organization.
Value single: M..64
Unit:
Example: Company VC1
Usage Notes: Put directions for finding additional information in the remark.

DUNS:
Contains the Dun & Bradstreet ID of the VC Provider organization.
Value single: M 8
Unit:
Example: 12-345-6789
Usage Notes: Needed for industry-wide unique VC and provider identification.

Address:
Describes the location of the VC provider organization.
Value single: M..256
Unit:
Example: 15243 Main St., Grovers Corners, NH 00000
Usage Notes:

Division:
Provides additional details concerning the specific division name or department within the organization, and the name of the primary individual for initial contact.
Value single: M..64
Unit:
Example: R & D
Usage Notes:

PrimaryContact:
Provides the name of the primary individual for initial contact.
Value single: M..64
Unit:
Example: Mick Mouse, Director
Usage Notes:

Phone:
Contains a primary contact phone number.
Value multiple: M..32
Unit:
Example: 1-1-408-111-1111
Usage Notes:

Email:
Contains the primary contact email address.
Value multiple: M..64
Unit:
Example: aaa@bbb.com
Usage Notes:

Fax:
Contains the primary contact fax number.
Value multiple: M..32
Unit:
Example: 1-1-408-222-2222
Usage Notes:

URL:
Contains the URL (Uniform Resource Locator) for finding additional information on the World Wide Web.
Value multiple: M..64
Unit:
Example: http://www.zzz.yyyy.com

Usage Notes:

2.2.31 Supplier Information: Transfer Package Information

VCT 1 Section 2.7.2

This information provides details about the transfer package including publication date(s), revision level, organization of the deliverables (media type and file structure), part numbers, and product nomenclature.

2.2.31.1 Package

This attribute provides information about the package of deliverables.

VCName:
Contains the name or alias of the packaged VC.
Value single: M..64
Unit:
Example: USB_device_101
Usage Notes:

VCID:
Contains a unique VC-identification number to be assigned by the VC provider.
Value single: M..64
Unit:
Example: M10-1234
Usage Notes: Must be unique in VC provider's product line.

2.2.31.2 PackageInformation

ArchiveFormat:
Describes the VC-deliverables archive format.
Value single: M..64
Unit:
Example: Tar
Usage Notes:

ArchiveDate:
Contains the date of authorship of the VC-deliverables package version.
Value single: N 8
Unit: yyyymmdd
Example: 19991108
Usage Notes: Begin at left starting with four digits for the year, followed by two digits for the month, and the remaining two digits for the day.

ArchiveVersion:
Contains the version information of the package release.
Value single: M..32
Unit:
Example: 001.023
Usage Notes: Must be unique for each VC package release. Should separate major versions from minor revisions by representing the version number to the left of the decimal point and the revision number to the right of the decimal point. For ease of sorting, the combined version-revision string should change monotonically throughout version history, for example, 002.004. (Note use of leading zeroes to prevent problems when more than nine releases occur.)

ArchiveMedia:
Describes the media used to transfer the package (for example, CD-ROM and Internet).
Value multiple: M..64
Unit:
Example: CD-ROM
Usage Notes: Explain in a remark how to obtain VC in each medium.

2.2.32 Supplier Information: Standard Terms and Conditions

VCT 1 Section 2.7.3

This section defines the VC provider's standard terms and conditions governing the evaluation and subsequent incorporation of the VC provider's intellectual property into the VC user's system(s).

2.2.32.1 SupportModels

This attribute provides information about the availability of support services from the VC provider.

Classification:
Classifies the support according to a standard list.
Value multiple: A..19
Unit:
Example: Training
Usage Notes: pick list
(Training | Maintenance | Design_Support | Application_Support)

2.2.32.2 FeeModels

This attribute provides fee information for use of the VC being described.

Classification:
Classifies the fee arrangements according to a standard list.
Value multiple: A..13
Unit:
Example: Royalty_Bearing
Usage Notes: pick list
(One_Time | Royalty_Bearing | Per_Usage_Fee | Subscription | Other)

2.2.32.3 ApplicablePatents

This attribute lists all applicable patents and pending patents filed by the VC provider and anyone else.

Description:
A free text list of patents applicable to the VC being described.
Value multiple: M..256
Unit:
Example: A Novel Method For Converting Analog Signals To Digital Data, US3387742, JP44976510
Usage Notes: Name/file number pairs can be concatenated in each entry.

2.2.33 Supplier Information: Third-Party Reference

VCT 1 Section 2.7.4

This section provides contact information for third parties, testing labs, and other partners who wish to be listed as suppliers of services to VC users.

2.2.33.1 Audits

This compound-type attribute identifies third-party organizations able to validate VC provider claims. Ideally, this validation is based on actual successful usage of the VC by the third party. However other levels of validation falling short of this standard may also be included.

Name:
Name of one organization.
Value single: M..64
Unit:
Example: XYZ Corporation
Usage Notes:

DUNS:
DUNS number for the named supplier.

Value single: M 11
Unit:
Example: 12-345-6789
Usage Notes:

Address:
A free text listing of address detail (such as street, city, region, postal code, and country, as appropriate).
Value single: M..64
Unit:
Example:
Usage Notes:

Division:
A single division and individual contact name.
Value single: M..64
Unit:
Example:
Usage Notes:

PrimaryContact:
The name of the primary individual for initial contact.
Value single: M..64
Unit:
Example: Mick Mouse, Director
Usage Notes:

Phone:
Telephone number, including country code.
Value multiple: M..32
Unit:
Example:
Usage Notes:

Fax:
Telephone number, including country code.
Value multiple: M..32
Unit:
Example:
Usage Notes:

Email:
Email address for named supplier.
Value multiple: M..64
Unit:
Example:
Usage Notes:

URL:
URL (including transport prefix) for named supplier.
Value multiple: M..64
Unit:
Example:
Usage Notes:

Maturity:
Classifies the level of audit information provided by the named third-party auditing organization according to a standard list.
Value single: M..15
Unit:
Example: Staged
Usage Notes: pick list (See “Note 3” in Appendix C “VCT2 Attribute Field/Value Pick List” for detailed explanation of list items.)

(Pre-registered | Registered | Staged | Experienced | Obsolete)

2.2.33.2 SupportingVCPartners

This compound-type attribute identifies third-party organizations willing to supply various VC services such as design, application, integration and validation.

Name:
Name of one organization.
Value single: M..64
Unit:
Example: Company A
Usage Notes:

DUNS:
DUNS number for the named supplier.
Value single: M 11
Unit:
Example: 23-456-7890
Usage Notes:

Address:
A free text listing of address detail (such as street, city, region, postal code, and country, as appropriate).
Value single: M..64
Unit:
Example:
Usage Notes:

Division:
A single division and individual contact name.
Value single: M..64
Unit:
Example:
Usage Notes:

PrimaryContact:
The name of the primary individual for initial contact.
Value single: M..64
Unit:
Example: Mick Mouse, Director
Usage Notes:

Phone:
Telephone number, including country code.
Value multiple: M..32
Unit:
Example: +1-408-555-1212
Usage Notes:

Fax:
Fax number, including country code.
Value multiple: M..32
Unit:
Example:
Usage Notes:

Email:
Email address for named supplier.
Value multiple: M..64
Unit:
Example:
Usage Notes:

URL:
URL (including transport prefix) for named supplier.
Value multiple: M..64
Unit:
Example:
Usage Notes:

ServiceType:
A free text listing of service type(s).
Value multiple: M..64
Unit:
Example: foundry
Usage Notes:

2.2.33.3 IndependentSoftwareVendors (ISVs)

This compound-type attribute lists information about third-party firms that supply software products applicable to VC usage.

Name:
Name of an organization.
Value single: M..64
Unit:
Example: Company B
Usage Notes:

DUNS:
DUNS number for the named supplier.
Value single: M 11
Unit:
Example: 98-765-4321
Usage Notes:

Address:
A free text listing of address detail (such as street, city, region, postal code, and country, as appropriate).
Value single: M..64
Unit:
Example:
Usage Notes:

Division:
A single division and individual contact name.
Value single: M..64
Unit:
Example:
Usage Notes:

PrimaryContact:
The name of the primary individual for initial contact.
Value single: M..64
Unit:
Example: Mick Mouse, Director
Usage Notes:

Phone:
Telephone number, including country code.
Value multiple: M..32
Unit:
Example: +33 (0) 555 55 21 21
Usage Notes:

Fax:
Fax number, including country code.

Value multiple: M..32
Unit:
Example:
Usage Notes:

Email:
Email address for named supplier.
Value multiple: M..64
Unit:
Example:
Usage Notes:

URL:
URL (including transport prefix) for named supplier.
Value multiple: M..64
Unit:
Example:
Usage Notes:

ProductType:
A free text listing of product type(s).
Value multiple: M..64
Unit:
Example: RTOS
Usage Notes:

2.2.34 Other: Customization

No corresponding VCT 1 section.

Use these attributes only when no other attribute is available for the type of information documented.

2.2.34.1 Custom

This compound-type attribute provides a means to define and describe a customized attribute.

Name:
Name of the custom attribute being defined.
Value single: M..64
Unit:
Example: packet_throughput
Usage Notes:

Description:
Free text custom attribute description.
Value single: M..256
Unit:
Example: packet processing throughput figures of the xyz network processor
Usage Notes:

Value:
Custom attribute value, conforming to the ISO format selected from VCT 2, Appendix E.
Value single: M..256
Unit:
Example: 64000
Usage Notes:

ISO_Format:
ISO format to which the custom attribute’s values must conform.
Value single: M..32
Unit:
Example: NR1..5
Usage Notes:

UnitOfMeasure:
Custom attribute unit of measure (as appropriate).
Value single: M..32
Unit:
Example: pps (packets per second)
Usage Notes:

3. Implementation Guidelines

3.1 VC Catalog Transfer

The efforts of the worldwide electronics industry to address design productivity issues are leading to the creation of more and more VC catalogs. As such catalogs become more available and comprehensive, VC users will increasingly utilize them to locate appropriate VC for their application purposes. Typically, proprietary catalogs evolve within the design organizations of corporate users to meet their specialized requirements, and public catalogs are produced by organizations serving the industry as information and/or product suppliers.

The need for a standard way to transfer VC information between the growing number of proprietary and open catalogs has been an important influence on the structure and content of the attributes designed for this standard. Care has been taken to define attributes and attribute fields that are independent of VC function and type. These definitions are intended to be comprehensive and generally applicable to all varieties of VCs. However, in cases where specific attributes do not apply to a particular VC, users of this standard are encouraged to implement just the subset of attributes pertinent to the VC being described. Note also that nothing in the design of attributes prohibits conducting transfers in a layered fashion based on the level of engagement and confidentiality established between the transferring parties.

By defining a common transfer format for VC catalog data, this standard serves the following purposes:

- Support the exchange of catalog data between multiple proprietary catalog sites (achieved by parsing or converting proprietary catalog data to and from VCT 2 format, as required).
- Establish a common set of catalog data items to be used by implementers of new VC catalogs.

Efficient transfer of VC catalog data is facilitated by following the semantics (meanings) and syntax (data types and expressions) defined by VCT 2. VC transfer is implemented by combining VCT 2 semantics and syntax with an appropriate transfer protocol that packetizes the data, and includes the necessary handshakes between the transferring parties to establish a reliable link. VCT 2 constrains only the attributes themselves and does not impose any limitations on the protocol selected or how the actual transfer is accomplished.

Figure 2: VC Transfer Implementation portrays the basic model for transfer implementation based on VCT 2. Note that VCT 2 only addresses the Presentation layer of the model. The other levels of the model are expanded upon in the following paragraphs.

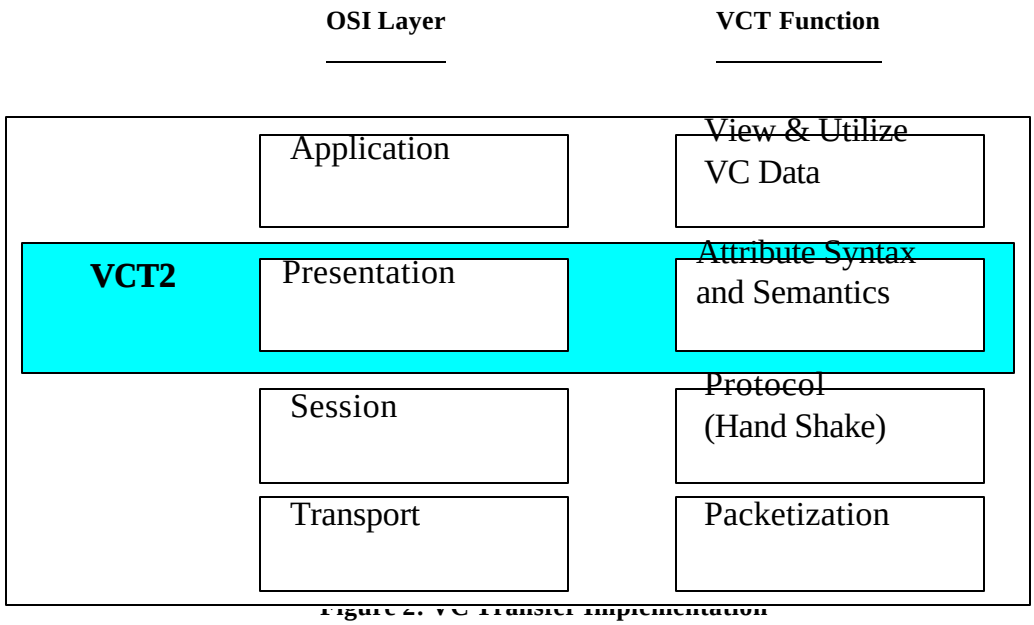


Figure 2: VC Transfer Implementation

Packetization Packetization gathers attributes into a packet of data for transfer. Transfer of VCT 2-compliant catalog data is accomplished by sending a stream of attribute fields and their values from the supplier to the consumer. A delimiter scheme that enables attribute fields to be distinguished from their values and one VC from the next in a catalog transfer is necessary for accurate and efficient transfer.

Protocol Protocol defines the procedure and sequence of events for communications between the sender and the receiver of VC catalog data. A set of commands and status messages is required.

Application The application enables VC catalog data users to effectively utilize the data for whatever design purposes are provided by the system. Application programs typically consist of a graphical user interface and interfaces to the protocol layer to initiate and terminate the transfer. As of the publication date of this standard, at least three different data-tagging schemes are known within and outside the electronics industry that may be appropriate for adaptation to VC transfer. Implementers are encouraged to consider these and other existing methods prior to undertaking a new design.

3.2 Usage Models

As described previously, the VCT 2 standard is designed to accommodate a wide range of VC types, usage models, and implementations. While the number of attributes used in any particular transaction (transfer scheme) will vary, attribute syntax and semantics apply to all usage models.

The following examples illustrate three likely usage models for the VCT 2 standard.

3.2.1 Company-to-Company Exchange Model:

This model highlights the use of VCT attribute formats as the common intermediate format between independent companies, each building and maintaining proprietary databases. Each company utilizes VCT 2 standard as a secondary format and translates their data into or out of it as needed. While the translation program is custom for each, once it is designed and debugged, it is reusable at the push of a button. Therefore each company is able to access the broadest industry availability of VC data to keep its internal proprietary information up to date and state of the art. (See Figure 3: Company-to-Company VC Exchange.)

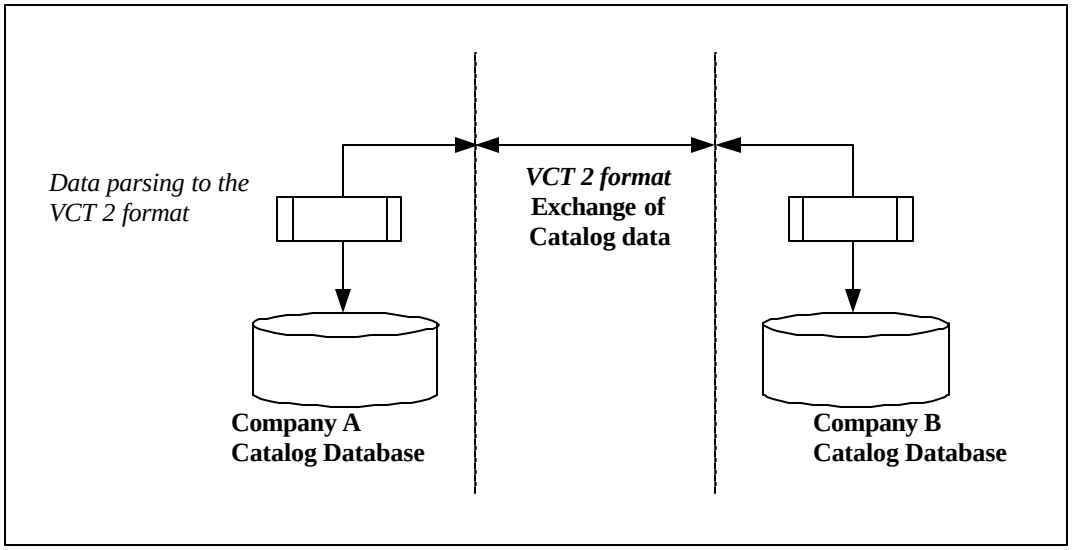


Figure 3: Company-to-Company VC Exchange

3.2.2 Broadcast Distribution Model

VC providers design their VC information to conform to VCT attributes either by originally authoring the data according to VCT 2 standard or by building the translator as part of the design process. All information is provided to their customers and partners using the VCT 2 representation. VC users download the data elements they require from published websites or contract for specific transfers as appropriate.

The acceptance of VCT 2 format by both VC providers and VC users enables VC providers to author the transfer data only once. All VC users are able to accept the data based on the pre-defined common format. (See Figure 4: VC Broadcast Distribution.)

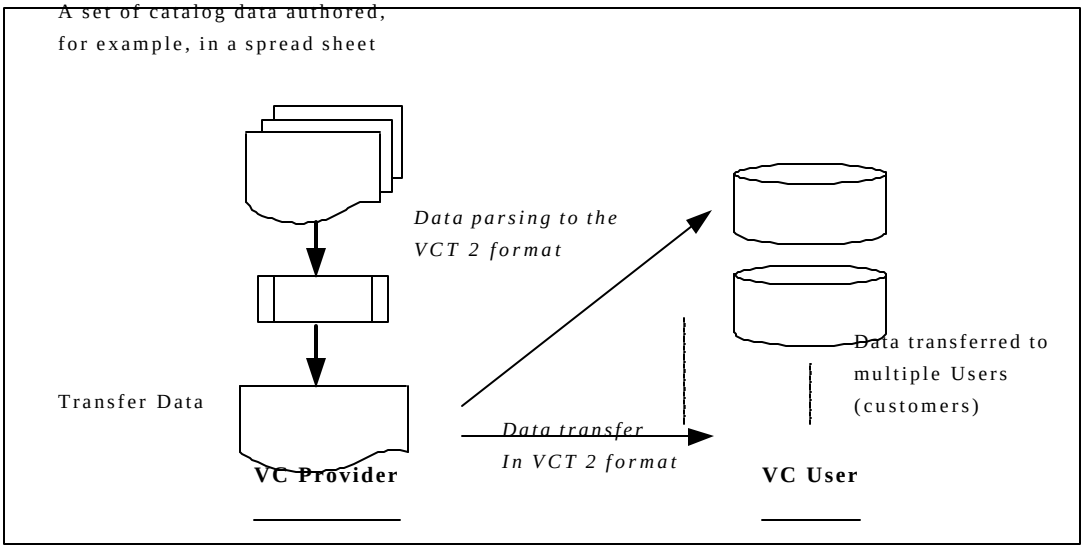


Figure 4: VC Broadcast Distribution

3.2.3 Query/Response Model

Figure 5: Query/Response VC Exchange shows the use of VCT 2 standard for VC search and selection using a query/response paradigm. In this example, the system designer seeks an appropriate VC for a system-on-chip design by sending queries structured in VCT 2 format to multiple catalog database sites. Upon receiving a query, each VC provider responds with answers in the same format (performing the translation from an internal proprietary format if necessary). Using the common semantics and syntax of VCT 2 not only makes the VC user's evaluation task easier, it increases the likelihood that the VC user will broaden the search and consider more alternate sources.

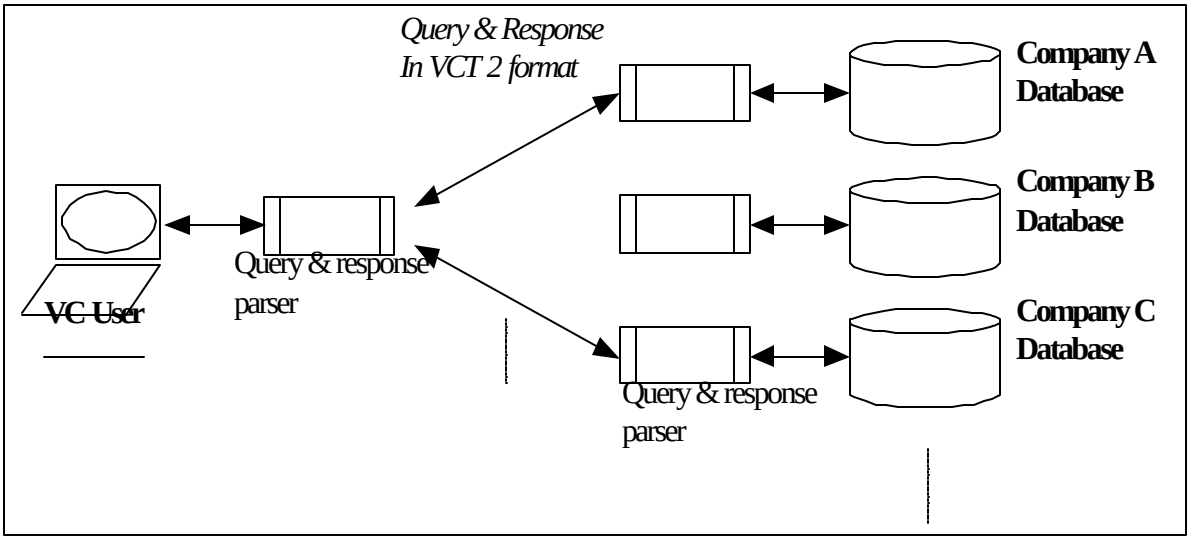


Figure 5: Query/Response VC Exchange

3.3 Attributes Guidelines

This section offers examples and illustrations designed to aid implementers in better conforming to VCT 2 attribute standard.

In VCT 2, each attribute field has exactly one value type defined. When VC data is transferred, the data will be sent as a sequence of attribute field and value pairs. In order to uniquely identify each attribute field, it is recommended that the descriptor and attribute names be concatenated with the attribute field name in the transferred data stream. Figure 6 provides several examples showing the name construction of attribute fields and illustrates how they are paired with their associated values.

Attribute Field Name	Attribute Field Value
ClassDescription	16Mb FLASH with dual write port
ClassClassification	Memory Elements/Nonvolatile/FLASH
ArchitectureDescription	NAND structure
⋮	
LatencySec	5.000E-8
⋮	
⋮	

Figure 6: Naming Attribute Field/Value Pairs

3.3.1 Flattening Methodology

Industry is using information such as defined herein in automated data interchanges. Flattening the hierarchy is required for some of these methodologies. VCT recommends that the period "." be used as the delimiter when concatenation of attributes with attribute fields is required.

3.3.2 Compound-Type Attributes

The Compound-type attribute is designed to accommodate the expression of two dimensional tables, which are often useful for documenting VC characteristics. VCT 2 utilizes the compound-type attribute when a table representation is the most appropriate way to organize specific VC information.

Figure 7: Examples of Compound-Type Attributes shows how the transferred data stream might appear when transferring VC deliverable information. Notice that the Attribute fields ("Name," "Comply," "Format," "FileName," and "Description") are repeated for each separate deliverable being transferred and may be repeated as often as necessary to transfer the complete set of deliverable information. Figure 7 also illustrates the vector nature of compound type attributes. The Descriptor "ListofDeliverablesDeliverable" (VCT 2 2.6) can be viewed as the vector name and each attribute field represents one element of the vector (each new deliverable value is equivalent to the next vector element).

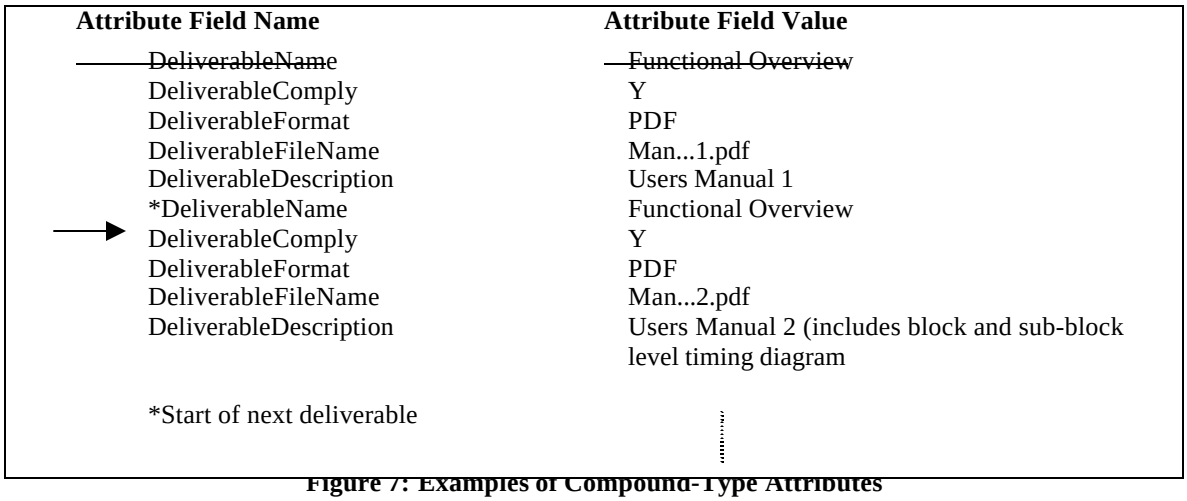


Figure 7: Examples of Compound-Type Attributes

3.3.3 Single- and Multiple-Type Attribute Fields

Each attribute field must be paired with one and only one value. However, a number of VC attributes (such as VCProviderContactInformation (VCT 2 Section 2.2.30.1) shown in Figure 8, may have multiple values that the VC provider desires to publish. Multiple values are accomplished by repetitively pairing the attribute field name with the next attribute field value until all paired values have been transferred. Such attribute field/value pairs are not compound because they have no determinate relationship with any other specific attribute field values. VCT 2 imposes no limitation on the number of values that may be assigned to a multiple-type attribute field.

Figure 8: Example of Single- and Multiple-Type Attribute Fields

3.3.4 Attribute Field Value Types

The attribute field value types in this document have been defined to make construction of efficient search criteria as easy as possible. Where possible, a pick list of standard values is provided. Detailed functional and market segment classification schemes are provided in Appendix A and Appendix B for use with the descriptors addressing these areas. Numerical data types have been defined using the ISO 6093 standard (Appendix E) to facilitate numeric value comparisons. For all attributes that fall outside of these types, the size of the free text field allocated for each attribute field is given. For such fields, please provide the most succinct and simple description possible.

3.3.5 Remark Capabilities

Remark attribute fields may be included as additional attribute fields under any attribute to provide pertinent supplementary information. Figure 9: Example of Remark Attribute Field, shows an example of the use of the Remark field in the PerformanceFrequency (VCT 2, Section 2.2.3.1) attribute category.

Frequency:Minimum	1.000000000000E06
Frequency:Typical	2.000000000000E06
Frequency:Maximum	3.000000000000E06
Frequency:Remark	figures for the specified throughput

Figure 9: Example of Remark Attribute Field

4. Acronyms and Abbreviations

The following is a list of acronyms and abbreviations:

BIST	Built-in Self Test
bps	bits per second
CAN	Controller Area Network
DFT	Design for Test
DMA	Direct Memory Access
DVD	Digital Versatile Disk
FIFO	first-in, first-out
FSM	Finite State Machine
Gbps	Gigabits per second
ICE	In-Circuit Emulator
IDDQ	I_{dd} Quiescent
ISV	Independent Software Vendor
ITU	International Telecommunication Union
Mbps	Megabytes per second
MIPS	Million instructions per second
OCB	On-chip Bus
PCI	Peripheral Component Interconnect
RTL	Register Transfer Level
RTOS	Real-Time Operating System
Sps	Samples per second
STIL	Standard Test Interface Language
UART	Universal Asynchronous Receiver Transmitter
URL	Uniform Resource Locator

5. Appendices

Several appendices are provided to give further motivation and explanation of the material in this document. The purpose for inclusion of these appendices is as follows:

A: VCT Functional Classification
Describes the VCT functional classification, which is used for assigning consistent functional categories to VCs being documented according to the VCT 2 standard.

B: VCT Market Segment Classification:
Describes the VCT market segment classification, which is used for assigning consistent market segment categories to VCs being documented according to the VCT standard.

C: VCT 2 Attribute Field/Value Pick List
Shows related pick lists in tabular form.

D: VCT 2 Implementer's Table
Represents the numeric and text values for attribute fields using the International Standards Organization (ISO) 6093 standard.

E: ISO 6093 Data-Type Definitions
References the ISO 6093 Data-Type Definitions used.

A. VCT Functional Classification

A.1 Introduction

This appendix describes the VCT functional classification, used for assigning consistent functional categories to VCs being documented according to the VCT 2 standard. This classification scheme was developed based on the following principles:

- Keep to a maximum of four levels of hierarchy.
- To avoid too many classes at any particular level, keep to a maximum of ten.
- The functional structure is the foundation.
- The application structure is mapped over the functional structure.

A.2 Scope

The VCT Functional Classification scheme facilitates the search process by providing a framework for the consistent definition of values for the functional class attribute (ClassClassification). The specific choices for each name used in this classification scheme were arrived at after a careful examination of functional taxonomies from Cadence, Design & Reuse, IP Highway, RAPID, and Toshiba. There is no limitation mandate as to the use of the VCT Functional Classification. A VC can be attached to more than one functional class if necessary for complete coverage of its possible functions.

The top and second levels contain an Others class for use in cases where the specific VC does not readily fit into any existing named category. As industry usage warrants, and it becomes appropriate to expand the classification scheme, new functional classes will be added to accommodate popular new functions.

A.3 Class Definitions

This section explains what each class name covers and defines the products covered by that class name.

Physical Library: The class of building blocks or elements used to assemble or compile a VC into a particular target process. The library provides a physical representation of the logic and functional elements of the design.

Memory Element: Any storage element from circular buffers and memory cells up to complete memory blocks.

Analog and Mixed Signal: A block or VC that includes an analog section requiring custom layout.

Arithmetic, Mathematic, and Logic Function: Basic digital functional blocks performing logic or mathematic functions; includes programmable logic.

Interface/Peripheral Core: Interfaces and peripherals that conform to recognized standards, or perform standard functions such as timers or keyboard controllers.

CODEC/Encryption: All blocks used to encode or encrypt information for transmission purposes.

Graphics/Image/Audio: Blocks that perform graphics-, audio-, or image-processing functions to recognized standards or provide functions such as Edge Detection.

Processors: Any VC or block that can run software to perform a task.

Subsystems: Blocks that are made by combining more than one subblock to form a subsystem or platform technology.

Test Functions: VCs that relate exclusively to test functions.

DSP Functions: VCs that perform digital signal processing functions. This category pertains to dedicated hardware blocks that perform specific DSP algorithms or routines.

Others: Any new functional class that arises in the future will be put in this category until a separate class is created.

Table 2: Class Definitions

Level 1	Level 2	Level 3	Level 4
Physical Library			
Memory Element	Portable Technology Optimized (for one specific process) I/O Macro Others		
Non-Volatile	RAM Cache Video RAM Static RAM Multiport RAM FlashRAM FRAM Building Blocks Others	DRAM 2/3/4/5 or more	 Asynchronous Synchronous
Analog and Mixed Signal	ROM EPROM PROM FLASH EEPROM Building Blocks/cells Others	 FIFO/LIFO Register Circular Buffer Queues	 Asynchronous Synchronous File Bank
	Data Acquisition		

Table 2: Class Definitions (Continued)

Level 1	Level 2	Level 3	Level 4
	D/A		
	RF		
	Signal Processing		
		Filters	
		Couplers	
		Doppler	
		Target and Clutter	
		Mixers	
		Multiplier/Divider	
	Comparators		
	Amplifiers		
	Detector		
	Pulse Compression		
	Sources		
	Switches		
	PLL/VCO		
	Reference/Regulator		
	Pulse Width Modulator		
	Others		
Arithmetic, Mathematic, and Logic Function			
	Comparator		
	Counter		
	Incrementer/Decrementer		
	Shifter		
	Multiplexer/Demultiplexer		
	Register/Buffer		
	Barrel Shifter		
	Adder		
	Multiplier		
	Delay Line		
	ALU		
	Bit Manipulation		
	Programmable Logic		
		FPGA	
		PAL	
		GAL	
		PLD	
		Others	
	Others		

Table 2: Class Definitions (Continued)

Level 1	Level 2	Level 3	Level 4
	Receiver/Transmitter		
		UART	
		Asynchronous	
		Synchronous	
		Others	
	Interface		
		XDSL	
		PCI	
		USB	
		PCMCIA	
		IEEE1284	
		IEEE1394	
		Physical Interfaces	
		ATM	
		IrDA	
		ISDN	
		T1/E1	
		SONET	
		Ethernet (IEEE802.x)	
		Bluetooth	
		Others	
	Timers		
	DMA Controller		
	Keyboard Controller		
	Others		
CODEC/Encryption:			
	Encoder/Decoder		
		JPEG	
		MPEG	
		ADPCM	
		AAC	
		AC-3	
		Viterbi	
		Reed Solomon	
		Others	
	Encryption/Decryption		
		DES	
		RSA	
		Others	
	Error Correction/Detection		
		FEC	
		Others	

Table 2: Class Definitions (Continued)

Level 1	Level 2	Level 3	Level 4
Graphics/Image/Audio	Others	QAM QPSK FSK OFDM Others	
Processors	Conversion Color Conversion Edge Detection RAMDAC VGA 3D Geometry Engine Rendering Engine LCD Controller Graphics Processors Speech Generator Speech Recognition Equalizer Others		
	Controller	Audio Memory Interrupt I/O Disk Cache Video Timer/Watchdog DMA Others	
	MCU	RISC	4/8/16/32/64/128 bit
		CISC	4/8/16/32/64/128 bit
	MPU	Others RISC	

Table 2: Class Definitions (Continued)

Level 1	Level 2	Level 3	Level 4
Subsystems	DSP (Processor Cores) VLIW SIMD MIMD Others	CISC Others Fixed Floating Core Others	32/64/128 bit 16/20/24 bit 16/24/32 bit 16/24/32 bit Audio Ethernet Others
Test Functions			
DSP Functions	Debug Self-Test Logic Memory Test Control Test Structures Others		
Others	Accelerator Building Blocks Correlator Filter Transformer Others		

B. VCT Market Segment Classification

B.1 Introduction

This appendix describes the VCT market segment classification, used for assigning consistent market segment categories to VCs being documented according to the VCT 2 standard.

B.2 Scope

The scheme for VCT market segment classification facilitates the search process. It provides a method for consistently defining values for the market segment class attribute field (MarketSegmentClassification). There is no limitation or mandate as to the use of this classification. A VC can be attached to more than one market segment class if necessary for complete coverage.

The top level contains an Others class for use in cases where the specific VC does not readily fit into any existing named category. As industry usage warrants, and it becomes appropriate to expand the classification scheme, new market segment classes will be added to accommodate emerging market segments.

B.2.1 Market Segment Class Names

Data Processing: Any product area involving the use of PCs and computer peripherals.

Communications: Voice and data communications, such as wireless and wired telephony, or networking products.

Consumer Electronics: Electronic products used in the home or for entertainment.

Automotive/Ground-based Transportation: All applications that are used for transport, apart from Aircraft. Includes all electronics used in cars such as engine management, ABS, and airbag systems.

Industrial and Medical: Industrial control and medical applications.

Military/Civil Aerospace: Aircraft and military products being generally more robust and higher level of quality than the other classes.

Others: Products that do not fit in to the above classes.

Table 3: Market Segment Classification

Level 1	Level 2	Level 3
Data Processing		
Communications	Servers PC Microprocessors PC Graphics PC Core Logic PC Communications Printers Scanners DRAM	
	Answering Machines ATM Cellular/PCS Telephones Hub Switches	

Table 3: Market Segment Classification (Continued)

Level 1	Level 2	Level 3
	Transceivers Line Cards	
Consumer Electronics	Modems SONET/SDH T/E Carrier Telephones ISDN Wireless Local Loop Wireless LAN	Analog xDSL Cable Corded Cordless
Automotive/Ground-based	Set Top Boxes (cable, satellite, terrestrial) Video Game Controllers DVD Video and Audio Video CS Digital Still Cameras Televisions (including DTV) VCRs Camcorders (analog and digital) CD Players Personal Portable Stereos (boom boxes, minidiscs) Stereo Components Multimedia PCs (top-level data only) Appliances	
Transportation Industrial and Medical	 In-car Entertainment Driver Information Vehicle/Body Control Power Train Safety and Convenience	
Military/Civil Aerospace		
Others		

C. VCT 2 Attribute Field/Value Pick List

Table 4: Attribute Field/Value Pick List

Descriptor Name	Attribute Name	Attribute Field	Pick List Values
Functional Overview	Class	Classification	Functional classification (Appendix A)
Target Applications	MarketSegment	Classification	Market segment classification (Appendix B)
Form Information	Hardness	Classification	(Soft | Firm | Hard | AMS)
List of Deliverables	Deliverable	Comply	(Y | N | Does_Not_Apply)
Abstract Models	Modeling-Abstractions	ModelType	(Partially_Ordered_Event_Accurate | System_Event_Accurate | Token_Cycle_Accurate | Instruction_Cycle_Accurate | Cycle_Approximate_Accurate | Cycle_Accurate | Gate_Propagation_Accurate)
Abstract Models	Modeling-Abstractions	ModelResolution-InternalTemporal	(Gate_Propagation | Clock_Accurate | Cycle_Approximate | Instruction_Cycle | Token_Cycle | System_Event | Partial_Order)
Abstract Models	Modeling-Abstractions	ModelResolution-InternalData	(Bit | Format | Value | Property | Token)
Abstract Models	Modeling-Abstractions	ModelResolution-InternalFunctional	(Digital_Logic | Algorithmic | Mathematical)
Abstract Models	Modeling-Abstractions	ModelResolution-InternalStructural	(Structural | Block_Diagram | Black_Box)
Abstract Models	Modeling-Abstractions	ModelResolution-ExternalTemporal	(Gate_Propagation | Clock_Accurate | Cycle_Approximate | Instruction_Cycle | Token_Cycle | System_Event | Partial_Order)
Abstract Models	Modeling-Abstractions	ModelResolution-ExternalData	(Bit | Format | Value | Property | Token)
Abstract Models	Modeling-Abstractions	ModelResolution-External-Functional	(Digital_Logic | Algorithmic | Mathematical)
Abstract Models	Modeling-Abstractions	ModelResolution-ExternalStructural	(Structural | Block_Diagram | Black_Box)
Abstract Models	Modeling-Abstractions	ModelResolution-Software-Programming	(Object_Code | Micro_Code | Assembly_Code | HLL_Statements | DSP_Primitive | Major_Modes | NULL)
Abstract Models	Modeling-Abstractions	ModelOf-Computation	(CSP | CT | DE | DT | PN | SDF | SR) **See Note 1 below.**
Structural Views	Internal-Observability-Structure	InternalVarField	(Undefined | File | Record | vsi_int8 | vsi_int16 | vsi_int32 | vsi_int64 | vsi_uint8 | vsi_uint16 | vsi_uint32 | vsi_uint64 | vsi_string | vsi_bit | vsi_mvl | vsi_bitvector | vsi_mvlvector | vsi_signed | vsi_unsigned | vsi_fxval | vsi_fxnum | vsi_Tbitvector | vsi_Tmvlvector | vsi_Tsigned | vsi_Tunsigned | vsi_fixed | vsi_ufixed)
System-Level Interfaces	PortDescriptions	PortTypeControl	(Null (No_Control) | Responder | Initiator)

Table 4: Attribute Field/Value Pick List (Continued)

Descriptor Name	Attribute Name	Attribute Field	Pick List Values
System-Level Interfaces	PortDescriptions	PortTypeData	(Null (No_Data) | Consumer | Producer)
System-Level Interfaces	PortDescriptions	TransactionType	(transRead | transWrite | messSense | messEmit | transOpenChannel | transCloseChannel | transSynchronize | transReset | transControl | messRead | messWrite)
System-Level Interfaces	PortDescriptions	PortFlow	(Persistent | Buffered | FIFO | LIFO | Blocking, Assigned_Port_Priority | Assigned_Data_Priority | Multi_Rate | Pipelined | Exceptions_Handled)
Logical (Mapped) Interfaces	Ports	ApplicationType	(Address | Data | Control | Status | Other)
Logical (Mapped) Interfaces	Ports	DataType	(Undefined | File | Record | vsi_int8 | vsi_int16 | vsi_int32 | vsi_int64 | vsi_uint8 | vsi_uint16 | vsi_uint32 | vsi_uint64 | vsi_string | vsi_bit | vsi_mvl | vsi_bitvector | vsi_mvlvector | vsi_signed | vsi_unsigned | vsi_fxval | vsi_fxnum | vsi_Tbitvector | vsi_Tmvlvector | vsi_Tsigned | vsi_Tunsigned | vsi_fixed | vsi_ufixed | enum_vsi_q_mode | enum_vsi_o_mode | enum_vsi_switch_mode | enum_vsi_ context_begin | enum_vsi_numrep | enum_vsi_fmt)
Logical (Mapped) Interfaces	Ports	Direction	(Input | Output | Bidirectional)
System/Logic Test Suite	TestSuites	SuiteResolution-ExternalData	(Bit | Format | Value | Property | Token)
System/Logic Test Suite	TestSuites	SuiteResolution-External-Functional	(Digital_Logic | Algorithmic | Mathematical)
System/Logic Test Suite	TestSuites	SuiteResolution-ExternalStructural	(Structural | Block_Diagram | Black_Box)
Clock Distribution	Clocks	Network	(Balanced_Tree | H_Tree | Grid | Other)
Clock Distribution	Clocks	ActiveEdge	(Rising_Edge | Falling_Edge | Both_Edges | High_Level | Low_Level)
Design Constraints	Abstract	Operations	(Rotate | Flip | Mirror)
Design Constraints	Reset	Type	(Synchronous | Asynchronous | Both | None)
Verification of Claims	QualificationLevel	Classification	(Architecture_Simulation | RTL_Verification | Gate_Verification | Formal_Verification | Static_Timing_Analysis | FPGA | Prototype | Production | Others) **See Note 2 below.**
Test Strategy	TestMethods	Classification	(Partial_Scan | Full_Scan | Boundary_Scan | BIST | IDDQ | Other)
Standard Terms and Conditions	SupportModel	Classification	(Training | Maintenance | Design_Support | Application_Support)
Standard Terms and Conditions	FeeModel	Classification	(One_time | Royalty_Bearing | Per_Usage_Fee | Subscription | Other)
Third-Party Reference	Audit	Maturity	(Pre-registered | Registered | Staged | Experienced | Obsolete) **See Note 3 below.**

Note 1

CSP	concurrent threads with rendezvous
CT	continuous-time modeling
DE	discrete-event systems
DT	discrete time (cycle driven)
PN	process networks
SDF	synchronous dataflow
SR	synchronous/reactive

Note 2

Architectural Simulation	VC is verified in behavior-level simulation.
RTL Verification	VC is verified at RTL-level simulation or formal verification.
Gate Simulation	VC is verified at gate-level simulation or formal verification.
FPGA	VC is verified with FPGA or similar devices.
Prototype	VC is prototyped and verified in physical silicon.
Production	VC is in production for commercial use.
Other	Use a text description if the values above are not applicable.

Note 3

Pre-registered	Auditing organization has collected some publicly available market data on the VC for internal research. Formal relationship with VC provider is not in place.
Registered	Auditing organization has published information from the VC provider including VC capabilities, manufacturing experience, design environment support requirements, and end-user licensing terms in a database available to potential VC users.
Staged	The VC is installed in-house at the auditing organization under configuration control and has been confirmed to be compatible with the tools, flows, and methodologies required by the responsible design group. Auditing organization is able to provide fully trained experts ready for engagement by potential VC users.
Experienced	The VC provider and the auditing organization have successfully taken to silicon one or more projects involving this VC.
Obsolete	VC is no longer available for new applications; a database of usage experience exists and is updated periodically from ongoing field installations for reference purposes.

D. VCT 2 Implementers's Table

To ease the task of implementing the details of VCT 2, the following information from Section 2.2 in this document is repeated in a convenient table.

Table 5: Implementer's Table

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
Class		Description	single	M..256	
		Classification	multiple	M..128	
Architecture		Description	single	M..256	
ModesOfOperations		Description	multiple	M..256	
EquivalentPartsOrCores		Description	multiple	M..256	
MarketSegments		Description	single	M..256	
		Classification	multiple	M..128	
Frequency		Minimum	single	NR3..1.12ES..2	hertz
		Typical	single	NR3..1.12ES..2	hertz
		Maximum	single	NR3..1.12ES..2	hertz
PowerConsumption		Minimum	single	NR3..1.4ES..2	watt
		Typical	single	NR3..1.4ES..2	watt
		Maximum	single	NR3..1.4ES..2	watt
Throughput		bps	single	NR3..1.3ES..2	bits per second
		fps	single	NR3..1.3ES..2	frames per second
		MIPS	single	NR3..1.3ES..2	million instructions per second
		sps	single	NR3..1.ES..2	samples per second
Latency		Mode	single	M..64	
		SystemLevel-Interface	single	M..64	
		clock	single	NR3..1.3ES..2	clock cycle
		sec	single	NR3..1.3ES..2	second
Hardness		Classification	single	A..4	
GateCount		Minimum	single	NR1..10	gate
		Typical	single	NR1..10	gate
		Maximum	single	NR1..10	gate
BitCount		Minimum	single	NR1..10	bit
		Typical	single	NR1..10	bit
		Maximum	single	NR1..10	bit
TotalArea		Minimum	single	NR3..1.3ES..2	mm^2
		Typical	single	NR3..1.3ES..2	mm^2
		Maximum	single	NR3..1.3ES..2	mm^2

Table 5: Implementer's Table (Continued)

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
		Typical	single	NR3..1.3ES..2	mm^2
		Maximum	single	NR3..1.3ES..2	mm^2
MemoryArea		Minimum	single	NR3..1.3ES..2	mm^2
		Typical	single	NR3..1.3ES..2	mm^2
		Maximum	single	NR3..1.3ES..2	mm^2
PinCount		Input	single	NR1..5	
		Output	single	NR1..5	
		Bidirectional	single	NR1..5	
		Test	single	NR1..5	
		Other	single	NR1..5	
FlipFlopCount	X	Value	single	NR1..10	
		ClockDomain	single	M..64	
StuckFaultCoverage	X	Name	single	M..64	
		Value	single	NR2..3.3	percent
CodeCoverage		Branch	single	NR2..3.3	percent
		Variable	single	NR2..3.3	percent
		Statement	single	NR2..3.3	percent
		FSM_arc	single	NR2..3.3	percent
		FSM_expression	single	NR2..3.3	percent
		FSM_state	single	NR2..3.3	percent
Deliverables	X	Name	single	M..256	
		Comply	single	A..12	
		Format	single	M..64	
		Filename	single	M..64	
		Description	single	M..256	
Availability		Date	single	N 8	yyyymmdd
		Reference	single	N 8	yyyymmdd
CompliantStandard	X	Organization	single	M..64	
		Name	single	M..64	
		Version	single	M..32	
		Revision	single	M..32	
Features		Description	multiple	M..256	
ModelingAbstractions		ModelName	multiple	M..256	
		ModelType	single	M..256	
		ModelResolution-InternalTemporal	single	A..17	
		ModelResolution-InternalData	single	A..8	
		ModelResolution-InternalFunctional	single	A..13	
		ModelResolution-InternalStructural	single	A..13	

Table 5: Implementer's Table (Continued)

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
		InternalTemporal ModelResolution-	single	M..17	
		ExternalData ModelResolution-	single	A..8	
		External-Functional ModelResolution-	single	A..13	
		ExternalStructural ModelResolution-	single	A..13	
		SoftwareProgram-ming LanguageSupport	single single	M..14 M..64	
		ModelOf-Computation ToolName	single single	A..3 M..256	
InterfaceStructure		InterfaceName	single	M..64	
		Interface-Description	single	M..256	
InternalObservability-Structure		InternalVarName InternalVarField	single single	M..64 M..64	
		InternalVar-Description	single	M..256	
PortDescription		PortName	single	M..64	
		InterfaceName	single	M..64	
		PortTypeControl	single	M..9	
		PortTypeData	single	M..9	
		TransactionType	single	M..16	
		PortFlow	multiple	M..22	
		Description	single	M..256	
LogicInterfaceType		Description	single	M..256	
Ports	X	Name	single	M..64	
		ApplicationType	single	A..7	
		DataType	single	M..32	
		Direction	single	A..13	
		Width	single	NR1..4	
		Description	single	M..256	
ConfigurationRegisters		Present	single	M..3	
		Description	single	M..256	
InterruptInterface		Name	single	M..64	
		Description	single	M..256	
		Type	single	M..64	
		InterruptPort	single	M..64	
ArbitrationInterface		Name	single	M..64	

Table 5: Implementer's Table (Continued)

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
		Type	single	M..64	
		ArbitrationPort	single	M..64	
MemoryAllocation		StartAddress	single	X..12	hexadecimal
		MinimumSize	single	NR3..1.14E..15	number byte
		MaximumSize	single	NR3..1.14E.15	byte
APISupport		APIName	single	M..64	
		APIStandard-Reference	single	M..256	
		APIDescription	single	M..256	
TestSuites		SuiteName	single	M..256	
		SuiteResolution-ExternalData	single	A..8	
		SuiteResolution-External-Functional	single	A..13	
		SuiteResolution-External-Structural	single	A..13	
Duty		Minimum	single	NR2..3.3	percent
		Typical	single	NR2..3.3	percent
		Maximum	single	NR2..3.3	percent
Period		Minimum	single	NR3..1.12ES..2	second
		Typical	single	NR3..1.12ES..2	second
		Maximum	single	NR3..1.12ES..2	second
Phase		Minimum	single	NR1..2	percent
		Typical	single	NR1..2	percent
		Maximum	single	NR1..2	percent
SupplyVoltage		Minimum	single	NR2..3.3	volt
		Typical	single	NR2..3.3	volt
		Maximum	single	NR2..3.3	volt
OperatingTemperature		Minimum	single	NR1 S..3	degree Celsius
		Typical	single	NR1 S..3	degree Celsius
		Maximum	single	NR1 S..3	degree Celsius
SNR	X	SignalName	single	M..64	
		Minimum	single	NR2 S..3.3	db
		Typical	single	NR2 S..3.3	db
		Maximum	single	NR2 S..3.3	db
Sensitivity	X	SignalName	single	M..64	
		Minimum	single	NR2 S..3.3	db
		Typical	single	NR2 S..3.3	db
		Maximum	single	NR2 S..3.3	db
I/O_Characteristic	X	Name	single	M..64	

Table 5: Implementers's Table (Continued)

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
OutputCharacteristic	X	SignalName	single	M..64	
		Metric	single	M..32	
		Value	single	NR3..3.3ES..2	
		UnitOfMeasure	single	M..32	
Clocks		Network	multiple	A..13	
		SkewRise	single	NR3..1.5ES..2	second
		SkewFall	single	NR3..1.5ES..2	second
		SlewRise	single	NR3..1.5ES..2	second
		SlewFall	single	NR3..1.5ES..2	second
		Jitter	single	NR2..3.3	percent of clock period
		ActiveEdges	multiple	A..12	
Abstract		Ratio_Min	single	NR2..2.2	
		Ratio_Max	single	NR2..2.2	
		Operations	multiple	A..6	
		Ratios_supported	multiple	NR2..2.2	
Reset		Types	multiple	A..12	
SpecifiedProcesses		Description	multiple	M..256	
SpecifiedFoundries		Description	multiple	M..256	
ProcessSensitivity	X	SignalName	single	M..64	
		Description	single	M..256	
ImplementationTestSuite	X	TestSuiteName	single	M..256	
		TestTime	single	NR2..8.2	second
		TestFrequency	single	NR3..1.12ES..2	hertz
		Vector Size	single	NR1..16	
		TesterType	single	M..64	
QualificationLevels		Classification	multiple	A..23	
Certifications		Description	multiple	M..256	
EDAtools	X	Name	single	M..64	
		Version	single	M..32	
SupportTools		Description	multiple	M..256	
ReferenceDesigns		Description	multiple	M..256	
Library	X	Name	single	M..64	
		Vendor	single	M..64	
ProvenProcesses		Description	multiple	M..256	
ProvenFoundries		Description	multiple	M..64	
Version		Number	single	M..32	
		Date	single	N 8	yyyymmdd
BugList		Description	multiple	M..256	
RelatedVCs		Description	multiple	M..256	
CustomizationOptions		Description	multiple	M..256	

Table 5: Implementer's Table (Continued)

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
TestModule	X	Format	single	M..64	
		Time	single	NR2..8.2	second
		Frequency	single	NR3..1.12ES..2	hertz
		Size	single	NR1..16	
		Type	single	M..64	
ContactInformation		Name	single	M..64	
		DUNS	single	M 11	
		Address	single	M..256	
		Division	single	M..64	
		PrimaryContact	single	M..64	
		Phone	multiple	M..32	
		Email	multiple	M..64	
		Fax	multiple	M..32	
		URL	multiple	M..64	
Package		VCName	single	M..64	
		VCID	single	M..64	
PackageInformation		ArchiveFormat	single	M..64	
		ArchiveDate	single	N 8	yyyymmdd
		ArchiveVersion	single	M..32	
		ArchiveMedia	multiple	M..64	
SupportModels		Classification	multiple	A..19	
FeeModels		Classification	multiple	A..13	
ApplicablePatents		Description	multiple	M..256	
Audits	X	Name	single	M..64	
		DUNS	single	M 11	
		Address	single	M..64	
		Division	single	M..64	
		PrimaryContact	single	M..64	
		Phone	multiple	M..32	
		Fax	multiple	M..32	
		Email	multiple	M..64	
		URL	multiple	M..64	
		Maturity	single	M..15	
SupportingVCPartners	X	Name	single	M..64	
		DUNS	single	M 11	
		Address	single	M..64	
		Division	single	M..64	
		PrimaryContact	single	M..64	
		Phone	multiple	M..32	
		Fax	multiple	M..32	
		Email	multiple	M..64	

Table 5: Implementer's Table (Continued)

Attribute Name	Cmpd	Attribute Field Name	Attribute Field Entries	ISO 6093 Data Type	Unit
		ServiceType	single	M..64	
IndependentSoftware-Vendors (ISVs)	X	Name	single	M..64	
		DUNS	single	M 11	
		Address	single	M..64	
		Division	single	M..64	
		PrimaryContact	single	M..64	
		Phone	single	M..32	
		Fax	single	M..32	
		Email	single	M..64	
		URL	single	M..64	
		ProductType	single	M..64	
Custom	X	Name	single	M..64	
		Description	single	M..256	
		Value	single	M..256	
		ISO_Format	single	M..32	
		UnitOfMeasure	single	M..32	
-	-	Remark	single	M..256	

Note: The Remark attribute field may be added to any attribute, and it may be either simple or compound.

E. ISO 6093 Data Type Definitions

VCT 2 represents the numeric and text values for attribute fields using the International Standards Organization (ISO) 6093 standard. This standard defines the representation of both quantitative and non-quantitative values in character strings for information exchange. While it also includes detailed specifications for multiple language support, specialized control characters, message headers, and numerous other features, the portion of interest for VCT 2 shall include only the specific English language formats shown below.

Non-quantitative data value format types:

A = alphabetic, letters only
N = numeric, digits only
X = alphanumeric (letters and numbers only)
M = mixed, all characters allowed

(A variable field length starts with two dots. A fixed field length starts with one space.)

Table 6: Data Type Definitions

Alphabetic	Numeric	Alphanumeric	Mixed	Description
A..3	N..3	X..3	M..3	Variable length, up to 3 characters
A 8	N 8	X 8	M 8	Fixed length, 8 characters
A..17	N..17	X..17	M..17	Variable length, up to 17 characters
A..35	N..35	X..35	M..35	Variable length, up to 35 characters

Quantitative data value format types:

NR1	=	integers
NR2	=	rational numbers with decimal-mark (real)
NR3	=	rational numbers with decimal-mark and exponent-mark (floating point)
S	=	signed (positive or negative)
.	=	decimal-mark
E	=	exponent-mark, base 10: (A)E(B) represents the value Ax10^B

Examples:

NR1..4	Positive integers (indicated by NR1) up to 4 digits in length
NR1 S..4	Positive or negative integers, up to 4 digits in length
NR2..3.5	Positive real (indicated by NR2), up to 3 digits in length to the left of the decimal point and up to 5 digits in length to the right of the decimal point
NR2 S..3.5	Positive or negative real, up to 3 digits in length to the left of the decimal point and up to 5 digits in length to the right of the decimal point
NR3..3.5	Floating point (indicated by NR3), positive real, up to 3 digits in length to the left of the decimal point and up to 5 digits in length to the right of the decimal point, with a signed 2-digit exponent
NR3 S..3.5	Same as floating point, but both positive and negative floating point mantissa
NR1 4	Integers exactly 4 digits long
NR1 S..4	Up to 4 integer digits plus an optional sign