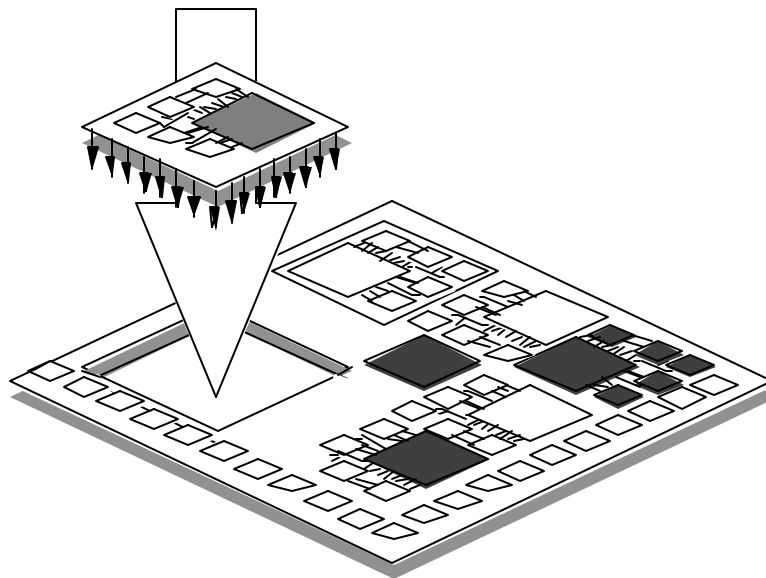


**VSI Alliance™**  
**Virtual Component Transfer**  
**Specification Version 2.1**  
**(VCT 1 2.1)**

**Virtual Component Transfer**  
**Development Working Group**

January 2001





## **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 (VCT 1 2.1)

### Company Members of the Development Working Group:

ARM	Cadence Design Systems
Design & Reuse	Escalade
Fujitsu	Hewlett-Packard Company
Infineon Technologies	NetLogic Microsystems
NOKIA Mobile Phones	Scottish Enterprises
Oki Electric Industry	Silicon Integration Initiative
STMicroelectronics	Synchronicity

### Individual Members of the Development Working Group

Lee Dilley  
Minoru Hasegawa  
Geeng-Wei Lee

### Contributors:

Patrick Cochenec .....	STMicroelectronics
Lee Dilley (Technical Writing Task Leader).....	Individual Member
Takeshi Fuse (Chairman).....	Fujitsu Ltd.
Anssi Haverinen.....	NOKIA
Katherine Hsu .....	Synchronicity
Al Kwok .....	NetLogic Microsystems
Ian Phillips .....	ARM Ltd.
Gabriele Saucier .....	Design & Reuse
John Teets, Melanie Yunk, Don Cottrell .....	Silicon Integration Initiative
Kumar Venkatramani.....	Cadence Design Systems
Mobashar Yazdani .....	Hewlett-Packard Company

### Editor

Sybil Sommer



## **Revision History**

Revision 2.1 - Editorial Staff, H. Leeds - Updated document to comply with Deliverables rules 23Oct00

Revision 2.1 - Editorial Staff - Copy edited and updated formats 26Jan01





## Table of Contents

<b>1. Overview .....</b>	<b>1</b>
1.1 Scope .....	1
1.2 Referenced Intellectual Property (IP) .....	1
1.3 Definition of Terms .....	1
1.4 Methodology Description .....	1
1.4.1 Design Process Centric View .....	1
1.4.2 Document Format .....	2
1.4.3 References to Other VSIA Specifications .....	2
1.5 Summary of Information Deliverables .....	2
1.6 References .....	6
<b>2. Specification of Documentation Deliverables .....</b>	<b>7</b>
2.1 VC Provider Claims .....	7
2.1.1 Functional Overview .....	7
2.1.2 Target Applications .....	7
2.1.3 Performance .....	7
2.1.4 Form Information .....	7
2.1.5 Test Coverage .....	7
2.1.6 List of Deliverables .....	7
2.1.7 Features and Standards Compliance .....	8
2.2 System /Logic Description .....	8
2.2.1 Functional Description .....	8
2.2.2 Abstract Models .....	8
2.2.3 Structural Views .....	9
2.2.4 Interfaces .....	9
2.2.5 Integration Requirements .....	10
2.2.6 System/Logic Test Suite .....	10
2.3 Physical Description .....	10
2.3.1 Interfaces .....	11
2.3.2 Integration Requirements .....	11
2.3.3 Implementation Test Suite .....	13
2.4 Reference Environment .....	13
2.4.1 Verification of Claims .....	14
2.4.2 Tools, Flows & Methodology .....	14
2.4.3 ASIC Libraries .....	14
2.4.4 Process Technology .....	14
2.4.5 Naming Convention .....	14
2.4.6 Deliverables Documentation .....	15
2.5 Application Information .....	15
2.5.1 Version History .....	15
2.5.2 Known Bugs .....	15
2.5.3 Application Notes .....	15
2.6 Test Information .....	16

2.6.1	Test Strategy . . . . .	16
2.6.2	Test Modules. . . . .	16
2.6.3	Test Modes . . . . .	17
2.6.4	Mixed Signal Test Integration . . . . .	17
2.7	Supplier Information . . . . .	18
2.7.1	VC Provider Contact Information. . . . .	18
2.7.2	Transfer Package Information . . . . .	18
2.7.3	Standard Terms and Conditions . . . . .	18
2.7.4	Third Party Reference. . . . .	18
<b>Appendix: Example of Structural Diagram. . . . .</b>		<b>19</b>

# 1. Overview

## 1.1 Scope

In the world of virtual component (VC) transfer (movement of a VC from provider/creator to user/integrator), it is essential that the proper set of information about the VC and its deliverables be supplied to users.

The objective of this document is to identify and specify the necessary and sufficient set of supplied information to facilitate the VC adoption process. This information set should enable users to correctly assess the applicability and practicality of adopting the VC. The set should also serve as a guide to the user for successful integration of the VC into his/her design.

This specification is not intended to standardize the documentation writing style, the order of sections presented, or any of the details of the documentation implementation.

Each item of the information set that requires documentation is presented in a high level description (with succinct explanation and brief examples) to facilitate the reader's understanding.

## 1.2 Referenced Intellectual Property (IP)

This document contains no specifically referenced IP.

## 1.3 Definition of Terms

<b>Platform</b>	The underlying enabling technology on which the object of reference is rendered functional. This can extend to the capabilities of engineering, support, and sales teams.
<b>Port</b>	Refers to a set of signals that reside on the boundary of a VC.
<b>VC Provider</b>	Refers to a person or an entity that originates and sources the VC in the VC transfer process (also known as VC creator).
<b>VC User</b>	Entity (also known as VC Integrator) that receives a VC in the Transfer process, as the counter-part to the VC provider.

## 1.4 Methodology Description

The structure of this specification is based on the following scheme:

### 1.4.1 Design Process Centric View

This document is structured from the viewpoint of the design process, which takes place when VC users consider the adoption or integration of a VC into their system. The design process consists of four steps: search, evaluation/selection, integration, and verification. A set of recommended information is supplied for each step.

Chapter 2 contains seven sections. VC Providers can assemble the information appropriate for their specific VC for each section. Figure 1 shows the section characteristics usually applicable for specific steps in the process.

**Figure 1: Design Process Step Characteristics**

Section No.	Section Title	VC Transfer Characteristic
2.1	VC Provider Claims	S, E
2.2	System/Logic Description	S, E, I, V
2.3	Physical Description	S, E, I, V
2.4	Reference Environment	S, E, I, V
2.5	Application Information	S, E, I
2.6	Test Information	S, E, I, V (manufacturing stage)
2.7	Supplier Information	S, E

S	Search
E	Evaluate/Select
I	Integration
V	Verification

This specification does not discuss VC functionality (choice of one type of VC over another, such as the choice of a processor VC over a phase locked loop VC). This specification details the set of elements a VC user needs to competently choose a specific VC to use in a System-On-Chip design.

#### 1.4.2 Document Format

The set of requirements specified is intended to include all information necessary for search, evaluation/selection, integration, and verification. The itemization is done in a modular fashion. The actual implementation, or format, of the documentation (e.g. user manual, design manual, data sheet, etc.) is left to the discretion of the VC provider. This documentation specification is descriptive as opposed to prescriptive.

#### 1.4.3 References to Other VSIA Specifications

When appropriate and relevant, references to the primary VSIA technical specification source documents, which provide definitive information about any specific addressed in this document, are included. (See Section 1.6.)

### 1.5 Summary of Information Deliverables

Table 1 uses the following codes to indicate how critical a deliverable is:

<b>M</b>	Mandatory
	A mandatory deliverable is required to make most chip designs workable that use the specified “hardness” of a VC indicated by the column it resides in.
<b>CM</b>	Conditionally mandatory (requirement is based on application)
<b>R</b>	Recommended
	A recommended deliverable will improve the design time, quality or accuracy for most chip designs that use the particular “hardness” of a VC indicated by the column it resides in.
<b>CR</b>	Conditionally recommended (requirement is based on application)
	For CM and CR, comments should identify a class of designs, VCs or chips containing VCs, where this deliverable is applicable if the defined condition is met. Conditions are sufficiently described to delineate the class of designs.

**Comments** Comments supply clarifying information. The specific conditions necessary to meet a conditionally mandatory or a conditionally recommended, are described.

**Table 1: VSIA Data Deliverables**

Section	Deliverable	VSIA Endorsed Formats	Soft	Firm	Hard	Comments
2.1	VC Provider Claims					
2.1.1	Functional Overview	ASCII, PDF, HTML	M	M	M	
2.1.2	Target Applications	ASCII, PDF, HTML	R	R	R	
2.1.3	Performance	ASCII, PDF, HTML	CM	CM	M	Provide for Soft & Firm if proven data from one or more implementation runs exists
2.1.4	Form Information	ASCII, PDF, HTML	M	M	M	
2.1.5	Test Coverage	ASCII, PDF, HTML	CM	CM	M	Provide for Soft & Firm if proven data from one or more implementation runs exists
2.1.6	List of Deliverables	ASCII, PDF, HTML	M	M	M	
2.1.7	Features and Standards Compliance	ASCII, PDF, HTML	M	M	M	
2.2	System Logic Description					
2.2.1	Functional Description	ASCII, PDF, HTML	M	M	M	
2.2.2	Abstract Models	ASCII, PDF, HTML	R	R	R	
2.2.3	Structural Views	ASCII, PDF, HTML	R	R	R	
2.2.4	Interfaces					
2.2.4.1	System- Level Interface	ASCII, PDF, HTML	M	M	M	
2.2.4.2	Logical (mapped) interfaces	ASCII, PDF, HTML	M	M	M	
2.2.5	Integration Requirements	ASCII, PDF, HTML	M	M	M	
2.2.6	System/Logic Test Suite	ASCII, PDF, HTML	M	R	R	

**Table 1: VSIA Data Deliverables (Continued)**

Section	Deliverable	VSIA Endorsed Formats	Soft	Firm	Hard	Comments
2.3	Physical Description					
2.3.1	Interfaces		--	--		
2.3.1.1	Timing Specifications	ASCII, PDF, HTML	CR	CM	M	Supply for Soft and Firm if either have been repeatably implemented and consistent measurements are available
2.3.1.2	Electrical Characteristics	ASCII, PDF, HTML	CR	CM	M	(same as above)
2.3.2	Integration Requirements					
2.3.2.1	Clock Distribution	ASCII, PDF, HTML	N/A	R	M	Does not apply to Soft VC
2.3.2.2	Design Constraints	ASCII, PDF, HTML	CM	R	M	Only Global signal constraints apply to Soft
2.3.2.3	Design Compatibility	ASCII, PDF, HTML	CM	CM	M	For Soft and Firm, mandatory only for Schematic symbol library and design database format
2.3.2.4	Process Compatibility	ASCII, PDF, HTML	CR	CM	M	Supply for soft if data known from previous implementation; supply for Firm if hard aspects of Firm are known
2.3.2.5	Process Requirements	ASCII, PDF, HTML	CR	CM	M	Supply for soft if data known from previous implementation; supply for Firm if hard aspects of Firm are known
2.3.2.6	Design Process Sensitivities	ASCII, PDF, HTML	CR	CM	M	Supply for soft if data known from previous implementation; supply for Firm if hard aspects of Firm are known
2.3.3	Implementation Test Suite	ASCII, PDF, HTML	M	M	M	
2.4	Reference Environment					
2.4.1	Verification of Claims	ASCII, PDF, HTML	M	M	M	

**Table 1: VSIA Data Deliverables (Continued)**

Section	Deliverable	VSIA Endorsed Formats	Soft	Firm	Hard	Comments
2.4.2	Tools, Flows & Methodology	ASCII, PDF, HTML	M	CM	CM	If a Tool, Flow, and/or Methodology applies to Firm and Hard, it must be supplied
2.4.3	ASIC Libraries	ASCII, PDF, HTML	CM	CM	CM	If ASIC Library has been used for any type (Soft, Firm or Hard) detail must be supplied
2.4.4	Process Technology	ASCII, PDF, HTML	CM	CM	M	If Soft and Firm have been implemented, information must be supplied
2.4.5	Naming Convention	ASCII, PDF, HTML	M	M	M	
2.4.6	Deliverables Documentation	ASCII, PDF, HTML	M	M	M	
2.5	Application Information					
2.5.1	Version History	ASCII, PDF, HTML	M	M	M	
2.5.2	Known Bugs	ASCII, PDF, HTML	M	M	M	
2.5.3	Application Notes		--	--		
2.5.3.1	Frequently Asked Questions	ASCII, PDF, HTML	R	R	R	
2.5.3.2	Design for Different Goals	ASCII, PDF, HTML	R	R	R	
2.5.3.3	Related/Family VC	ASCII, PDF, HTML	R	R	R	
2.5.3.4	Customization Options	ASCII, PDF, HTML	R	R	R	
2.5.3.5	Example Systems	ASCII, PDF, HTML	R	R	R	
2.6	Test Information					
2.6.1	Test Strategy	ASCII, PDF, HTML	CM	CM	M	For Soft and Firm strategy is applicable and should be supplied if equivalent to a prior existing implementation
2.6.2	Test Modules	ASCII, PDF, HTML	CM	CM	M	(same as Test Strategy)

**Table 1: VSIA Data Deliverables (Continued)**

Section	Deliverable	VSIA Endorsed Formats	Soft	Firm	Hard	Comments
2.6.3	Test Modes	ASCII, PDF, HTML	M	M	M	
2.6.4	Mixed Signal Test Integration	ASCII, PDF, HTML	--	--	--	
2.6.4.1	Signal Requirements for Analog Blocks	ASCII, PDF, HTML	N/A	N/A	CM	Not applicable to Soft or Firm and only applicable to Hard analog mixed signal VCs
2.6.4.2	Analog Test Requirements	ASCII, PDF, HTML	N/A	N/A	CM	Not applicable to Soft or Firm and only applicable to Hard analog mixed signal VCs
2.7	Supplier Information					
2.7.1	VC Provider Contact Information	ASCII, PDF, HTML	M	M	M	
2.7.2	Transfer Package Information	ASCII, PDF, HTML	M	M	M	
2.7.3	Standard Terms and Conditions	ASCII, PDF, HTML	M	M	M	
2.7.4	Third Party Reference		--	--	--	
2.7.4.1	Supporting VC Partner	ASCII, PDF, HTML	R	R	R	
2.7.4.2	ISV (Independent Software Vendor)	ASCII, PDF, HTML	R	R	R	

## 1.6 References

The following published VSIA specifications are referred to in this document.

**Note:** The primary VSIA document for referencing all other VSIA specification documents is the *VSIA Deliverables Document*. This document contains the most current version numbers.

Reference 1: *Test Data Interchange Formats and Guidelines for VC Providers Specification*

Reference 2: *Soft and Hard VC Structural, Performance and Physical Modeling Specification*

Reference 3: *Analog/Mixed -Signal VSIA Extension Specification*

Reference 4: *On-Chip Bus Attributes Specification*

Reference 5: *VSIA System-Level Design Model Taxonomy*

Reference 6: *System-Level Interface Behavioral Documentation Standard (to be released Q4 1999)*



## 2. Specification of Documentation Deliverables

### 2.1 VC Provider Claims

This section contains the VC provider's overview claims for the functionality, nature and completion of the specific VC. Since a VC is “virtual” by definition, provision of achieved (target) attributes of the VC, after it is realized for silicon, facilitates the assessment and understanding of the VC in the selection process.

#### 2.1.1 Functional Overview

The functional overview should summarize the key functions of the VC in a concise, compact style. This approach will enable potential VC users, in the search, evaluate/select mode, to rapidly determine the applicability of the VC being described. This deliverable is Mandatory (M).

#### 2.1.2 Target Applications

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. This deliverable is Recommended (R).

#### 2.1.3 Performance

Performance claims enable the VC user to determine if the VC is capable of meeting the overall performance expectations required for the VC user's design. Examples of performance claims include:

- Frequency of operation, clock (measured or estimated)
- Normalized measure(s) (MIPs, bps, etc. - measured or estimated)
- Power consumption (measured or estimated)
- Signal-to-noise ratio (SNR), gain, etc. (mixed signal VCs)
- Architectural performance numbers (e.g. bit rate error ratio)

This deliverable is Mandatory (M) for hard VCs. Soft and Firm VCs are Conditional Mandatory (CM). If there is valid data from one or more implementation runs for these two deliverables, they become Mandatory (M).

#### 2.1.4 Form Information

This section describes the hardness of the VC (soft, firm, or hard). A description of the estimated or exact gate count, area, pin counts (input, output, test), and dimensions are included. If software is part of the VC, indicate memory requirements for the compiled objects. This deliverable is Mandatory (M).

#### 2.1.5 Test Coverage

This section describes the quality of the verification test bench and manufacturing test bench in terms of code coverage, stuck fault coverage, etc., as appropriate for the technology of the VC. This deliverable is Mandatory (M) for hard VCs. Soft and Firm VCs are Conditional Mandatory (CM). If there is valid data from one or more implementation runs for these two deliverables, they become Mandatory (M).

#### 2.1.6 List of Deliverables

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

A summary of list of deliverables, in the form of one or more tables, is recommended. For example, see Table 1: Design Kit Deliverables in the *VSIA Deliverables Document*.

This deliverable is Mandatory (M).

### 2.1.7 Features and Standards Compliance

This section lists the main features and benefits of the VC, states the standards to which the VC is compliant, and indicates the specific revision of each standard applicable to the VC. The following example shows how a particular standard may be referenced with the corresponding applicable version. This deliverable is Mandatory (M).

Example: The Ethernet interface is compliant with IEEE 802.3.

## 2.2 System /Logic Description

This section contains the external description of how the VC functions and the nature of the VC interfaces on the system and logic level. The system/logic description is written for the VC user (system designer or VC integrator) to gain an understanding of the functional operation and general applicability of the VC. This section contains both implementation and operational information, and can range in complexity from a page or two for simple functions to hundreds of pages for complex microprocessors. If the VC has multiple implementation paths, tradeoff considerations should be included to assist the VC User in assessing applicability to his/her particular design. The system/logic description should convey the philosophy and design concept of the VC, complemented by the timing diagrams, etc. described in the following sections.

### 2.2.1 Functional Description

This section describes the general functionality of the VC. Detailed behavior of the VC should be provided. The details should include the specific functional architecture (IEEE802 standard for communication VC, Instruction Set Architecture for Processor VC, etc.), different modes of operation, specific features provided, etc.

The format and details of this section may vary in complexity and length depending on the functional complexity and the depth of understanding required for different types of VC. Sufficient unambiguous information should be supplied and specified to enable the VC user to understand the behavior of the VC in his/her application. This deliverable is Mandatory (M).

### 2.2.2 Abstract Models

This section contains information about

- models
- their abstraction level (Behavioral, Functional, Structural, etc.), as found in Section 1.6 [reference 5]
- how models are used in the design flow
- their accuracy (cycle vs. instruction), etc.

For example, this VC is supplied with a system evaluation model in C (C++), a Cossap (SPW) virtual prototype model implemented with floating point (fixed point) accuracy, and a Verilog bus functional model.

The availability of register transfer logic (RTL) models, whether these models are behavioral or structural, and what compatibility (if any) these models have with various synthesis tools needs to be indicated.

Descriptions of abstract models are important at the system/logic level both for the evaluation/selection stage and for the design stage.

This deliverable is Recommended (R).

### 2.2.3 Structural Views

This section provides the architectural description of the VC in a graphical manner. Structural diagrams provide an internal view of the VC and graphically describe the major blocks in a design and the dataflow between them. Information about the interface scheme may be included in the structural diagram to describe the communication among blocks at each level of abstraction. These diagrams are used to describe system functionality from a high level to the detail level. A schematic diagram is a block diagram with low level (circuit level) details added. This section supplements the Functional Description Section by facilitating the VC user's understanding of the implementation of the VC in addition to its behavior. See Appendix: Example of a Hierarchical Structural Diagram in this specification. This deliverable is Recommended (R).

### 2.2.4 Interfaces

This section describes the intended requirements and behavior of the VC on the block boundary at the system/logic level. The set of information supplied should assist the VC users to properly interface the VC to other blocks in a design and enable accurate and rapid selection of the VC if it is rendered as a black box.

#### 2.2.4.1 System-Level Interface

The system-level interface resides on the block boundary of abstract VC models. Adequate information about the system-level interface needs to be detailed for the VC user to properly understand the communication scheme of the abstraction models. The system-level interface is often constructed with “layers” of communication abstraction. These layers depend on the objectives of the VC model and the stage of system-level design in which the models are utilized. Sufficient information and instruction on communication layers should be detailed in this section. The link between layers of interface abstractions must also be clearly indicated. These links show the inheritance of communication properties across the abstraction layers. For further details of the system-level interface, see Section 1.6 [reference 6]. This deliverable is Mandatory (M).

#### 2.2.4.2 Logical (mapped) Interfaces

This section describes the attributes of a VC interface when it is mapped to register transfer level (RTL) or below. The information provided should be sufficient to enable integration of the VC with other VCs in a design. One or more of the following sections need to be provided.

See Section 1.6 [reference 4] for documentation of on-chip bus interfaces.

#### Structural Information

##### Register Interface

This section includes descriptions of any required configuration registers and memory mapping. Register interface descriptions should incorporate at least the following information:

- Read/write attributes
- I/O and bit map configurations

##### Port Descriptions

This section describes port names and their associated functions and signal directions. The ports are preferably grouped according to function (e.g. bus). Some examples of types of signals to be described are:

- Address and data
- Control signals
- Status signals

#### Behavioral Information

##### Timing Diagrams

Timing diagrams portray step by step sequential VC behavior and greatly facilitate the understanding of logical interface requirements. Timing diagrams graphically illustrate the clocks, data interfaces, status, etc. They identify valid conditions for the inputs and outputs (e.g., on which edge the input is captured and on which edge the output changes). VC Providers should include detailed information covering false path designations, timing errors that can be ignored, and set-up and hold conditions as appropriate.

### **Protocol**

Interface protocols, such as for bus interfaces, should be described in this section. The types of operations described are dependent upon the specific nature of the VC and may be expected to include one or more of the following (as well as any other operation types deemed pertinent):

- Protocol fundamentals (word formats, etc.)
- Arbitration
- Error handling
- Interrupts
- DMA data transfer

### **Data Format Information**

This is a specification of the data formats passed across the interface appropriate to the specific VC.

### **Software Information**

API (Application Program Interface): If the VC has control firmware embedded inside the boundary (e.g., VC includes onboard ROM for firmware), software requirements and instructions to interface this VC should be provided.

This deliverable is Mandatory (M).

## **2.2.5 Integration Requirements**

This section describes implementation platform requirements that are not adequately covered elsewhere. These include minimum processing power required of the embedded computing engine to reproduce VC claims, or particular design skills required of the implementation team. This section does not address design and layout sensitivities, which are covered in Electrical Characteristics (Section 2.3.1.2.)

Types of information documented in this section include:

- CPU Technology: e.g., 20 MIPS RISC with C++ Compiler and RTOS, etc.
- Implementation Capability: e.g., Radio Frequency (RF), Differential Logic, Hand-crafted DSP, etc.
- Logic Library: e.g., Acceptable sources, minimum performance specifications, etc.
- Application Expertise: e.g., Ability to support GSM, DVD, etc.
- Interfaces: e.g., non-standard protocols and/or formats, etc.

This deliverable is Mandatory (M).

## **2.2.6 System/Logic Test Suite**

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 instructions for utilizing any verification test bench provided. This deliverable is Mandatory (M) for Soft VCs, and Recommended (R) for Firm and Hard VCs.

## **2.3 Physical Description**

This section contains the description of the VC on the physical level whether the VC is supplied in a hard VC format or if the VC is intended to conform to certain physical characteristics when implemented. This section contains both implementation and operational information. If the VC has multiple implementation paths, discussion of any tradeoff considerations should be included to assist the VC User in assessing applicability to his/her particular design.

## 2.3.1 Interfaces

This section describes physical characteristics of the signals on the VC block boundary.

### 2.3.1.1 Timing Specifications

The physical level timing specifications should describe the requirements and behavior of the VC in terms of numerical time values. This timing specification, commonly known as the “AC Specification,” describes the dynamic characteristics of the VC.

Examples include:

- Clock frequency and duty cycle
- Reset pulse width requirement
- Address input set up time to the capture clock edge
- Requirements on the asynchronous inputs

See Section 1.6 [reference 2].

This deliverable is Mandatory (M) for Hard VCs. Soft VCs are Conditional Recommended (CR) and Firm VCs are Conditional Mandatory (CM). If the Soft and Firm VCs have been implemented repeatedly and consistent measurements are available, then the Soft and Firm VCs respectively become Recommended (R) and Mandatory (M).

### 2.3.1.2 Electrical Characteristics

The electrical specification of a VC should include a list of parameters (e.g., integral non-linearity (INL), differential non-linearity (DNL), gain, bandwidth) with the definition and calculation method for each.

These parameters include:

- Variation and/or sensitivity over temperature, and supply voltage
- Absolute maximum ratings
- Recommended operating conditions
- Electrical input and/or output specification
- Typical, minimum and maximum values
- Peak, typical, and minimum power drain with worst-case rate of change for each
- Number of simultaneous switching devices (NSSD)

This deliverable is Mandatory (M) for Hard VCs. Soft VCs are Conditional Recommended (CR) and Firm VCs are Conditional Mandatory (CM). If the Soft and Firm VCs have been implemented repeatedly and consistent measurements are available, then the Soft and Firm VCs respectively become Recommended (R) and Mandatory (M).

## 2.3.2 Integration Requirements

Integration requirements define the criteria for embedding the VC into the host design to facilitate plug-and-play capability.

### 2.3.2.1 Clock Distribution

All critical information on clocks and clock distribution must be supplied by the VC provider; all items to include methods of and conditions for specification.

Hard VCs must specify:

- Interface requirements for the clocks including frequency range, and duty cycle
- Rise and fall edges, active edges, slew rate, and jitter
- Relationship to other clocks, and clock stop/start conditions

Firm VCs should specify:

- Detailed description of the clock distribution network
- Clear definitions and descriptions of additional clock frequencies or phases being developed in the VC from the clock I/O on the VC
- Internal clocking distribution, internal clock delay and skew

Soft VCs should not include the internal clocking structures (see Section 1.6 [reference 2]).

This deliverable is Mandatory (M) for Hard VCs and Recommended (R) for Firm VCs. This deliverable is not applicable to Soft VCs.

### 2.3.2.2 Design Constraints

Design constraints are specified for the VC as a whole, for hierarchical blocks within the VC, or for particular signals within the VC. Design constraints include one or more of the following:

- Global signals constraints (e.g. global asynchronous set/reset)
- Area constraints
- Physical implementation constraints
- Test constraints
- Environmental/operating conditions
- Power constraints (VC power consumption requirements)

It is recommended that this information be documented in a table format (see Section 1.6 [reference 2]).

This deliverable is Mandatory (M) for Hard VCs and Recommended (R) for Firm VCs. Soft VCs are Conditional Mandatory (CM), becoming Mandatory (M) only when Global signal constraints apply.

### 2.3.2.3 Design Compatibility

This section details all pertinent information concerning:

- Electrical design rules (including VC HSpice models and HSpice version used)
- Device types and (construction) structures
- Topological design (layout) rules
- Border rules (between the VC and the host design)
- Mask level generation rules (grid size, sizing, etc.)
- Schematic symbol library
- Design rules check (DRC)/layout versus schematic (LVS) rule files
- Design data base formats

This deliverable is Mandatory (M) for Hard VCs. Soft and Firm VCs are Conditional Mandatory (CM), becoming Mandatory (M) only for the Schematic Symbol Library and the Design Database Format.

### 2.3.2.4 Process Compatibility

Process compatibility sets forth VC provider evaluations of:

- Process integration flow
- Process simulation files
- Material composition (conductive and insulation layers)
- Key process modules

This deliverable is Mandatory (M) for Hard VCs. Soft VCs are Conditional Recommended (CR). Firm VCs are Conditional Mandatory (CM). Soft VCs become Recommended (R) if there is data from a previous implementation. Firm VCs become Mandatory (M) if the hard aspects of the Firm VC are known.

### 2.3.2.5 Process Requirements

Process requirements define the necessary process conditions (process compatibility issues) for the VC to be integrated into the host design. These include:

- Geometry and Process Features (quad metal, double poly, etc.)
- Foundry where the VC was processed
- Process integration flow and device (construction) structures
- Mask level generation rules

For example, process requirements might be stated as: fabrication technology (0.25 $\mu$  CMOS, or 1.0 $\mu$  SOS, or 0.5 $\mu$  bipolar) with low-noise, epitaxial, trench-isolation.

This deliverable is Mandatory (M) for Hard VCs. Soft VCs are Conditional Recommended (CR). Firm VCs are Conditional Mandatory (CM). Soft VCs become Recommended (R) if there is data from a previous implementation. Firm VCs become Mandatory (M) if the hard aspects of the Firm VC are known.

### 2.3.2.6 Design Process Sensitivities

Process sensitivities should describe:

- Process parameters that affect the functionality, performance, and other characteristics of the VC.
- Known sensitivities to processing and how performance varies with device's parameters.

Examples include the sensitivity of the circuit operation to capacitor oxide thickness, poly resistivity, etc. Such information allows an integrator to see how performance specifications vary with process changes (see Section 1.6 [reference 3]).

This deliverable is Mandatory (M) for Hard VCs. Soft VCs are Conditional Recommended (CR). Firm VCs are Conditional Mandatory (CM). Soft VCs become Recommended (R) if there is data from a previous implementation. Firm VCs become Mandatory (M) if the hard aspects of the Firm VC are known.

### 2.3.3 Implementation Test Suite

This section contains information pertaining to the proof of satisfactory integration and implementation (as distinct from manufacturing test, which pertains to confirmation of satisfactory reproduction).

Descriptions of strategies and implementations providing proof of satisfactory integration of the VC in physical form are recommended. A statement justifying the chosen approach should also be considered. Depending on the strategies chosen, this section may contain data files of stimuli and response data for use on a tester. Text documents and diagrams illustrating bench tests of critical parameters may also be included. Section 2.6 provides information about requirements test related information details. This deliverable is Mandatory (M).

## 2.4 Reference Environment

This section contains specific descriptions of the technology used in the VC development, necessary for its successful deployment. Assessments of the sensitivity of the successful deployment of the VC to the specific tools, flows, and technology are included. In all cases revision numbers of libraries and tools should be clearly identified.

To facilitate the evaluation/selection of a VC prior to purchase, it is the VC provider's responsibility to clearly define the context in which all features, characteristics and performance claims were established and validated. This context will be referred to in this document as the "reference environment." A well designed, complete reference environment should enable the VC User to accurately duplicate all results and verify all VC Provider claims. The reference environment should, at a minimum, include:

- a generic technology library (typically, one readily available in the industry)
- the design flow
- the design tools used
- the options, parameters, and all other information used to implement the VC in this environment.

### 2.4.1 Verification of Claims

In the case of hard VCs, verification of the VC provider's claims and assumptions is accomplished by actual measurement. Verification of soft and firm VCs is achieved by running the design through to completion against the reference environment defined in Section 2.4 for each of the claims presented in Sections 2.1.1 through 2.1.7. It is recommended that the VC provider show compliance of the VC to each specification represented by sections of this document, as prescribed in Appendix D of the most recent version of the VSIA Deliverables Document.

Supplementary verifications should be included wherever possible to provide additional avenues for the VC user to confirm the critical aspects of VC operation most pertinent to his/her application. References to third parties having successfully implemented the VC provider's VC, along with details of achieved performance parameters, size, power dissipation, etc. are particularly valuable. If available, results from any laboratory certifications that may have been performed should also be included. This deliverable is Mandatory (M).

### 2.4.2 Tools, Flows & Methodology

The VC provider will document the reference environment for the VC including a minimum of the following:

- Electronic design automation (EDA) tool environment (identifying version details)
- Release status of compatible ASIC reference libraries
- Sources and release status of any compilers and linkers required
- Itemization of any specific tools used for performance measurements (SNR, International Telecommunications Union (ITU) standards compliance, etc.)
- Boundary conditions (range of operation, scalability) over which the stated performance claims can be validated

The VC provider's claims and the associated reference environment shall serve to enable the VC integrator to evaluate the appropriateness of the VC provider's products to his/her application.

This deliverable is Mandatory (M) for Soft VCs. Firm and Hard VCs are Conditional Mandatory (CM). If a Tool, Flow, and/or Methodology applies to Firm and Hard VCs, they become Mandatory (M).

### 2.4.3 ASIC Libraries

This section contains the specific description of the cell-library needs of the VC. In particular, it will identify the logic functions needed and the critical timing/power/drive characteristics of those cells.

This deliverable is Conditional Mandatory (M). If an ASIC Library has been used for any of the VC types, it becomes Mandatory (M).

### 2.4.4 Process Technology

This section describes the technology with which the VC has been verified:

- Processing history/portability: The list of the processes into which the VC has been successfully integrated and the manufacturers' processes assumed to be compatible (e.g. BICMOS6 vs. HCMOS6).
- Process design rules exceptions: Although such exceptions should be avoided whenever possible, on occasion an analog design will violate design rules to achieve unique intellectual property (e.g., DRC violations in memories, recognition of certain components for LVS). These design rules exceptions must be clearly noted with a detailed explanation.

This deliverable is Mandatory (M) for Hard VCs. Soft and Firm VCs are Conditional Mandatory (CM). If Soft and Firm VCs have been implemented, they become Mandatory (M).

### 2.4.5 Naming Convention

The use of a naming convention is considered normal practice for VC and block designers. Such usage is especially needed to document intra-chip or intra-block conventions.

More stringent naming requirements are necessary when super assemblies of blocks are used to avoid naming clashes between blocks from different projects. An overall naming convention is essential in this case.



For VSIA naming rules, refer to Section 1.6 [reference 2].

This deliverable is Mandatory (M).

## 2.4.6 Deliverables Documentation

This section describes the documentation associated with the deliverables defined in the *VSIA Deliverables Document*. Types of items to be covered include:

- The installation guide describing how to adopt deliverables in a VC user's design environment.
- The usage guide describing the purpose of the deliverables and the proper usage of the deliverables in the design process.
- The conventions and criteria used (e.g. definition of layers of GDSII-Stream must be provided for a hard VC).
- A description of GDSII-Stream is found on the VSIA website, [www.vsi.org](http://www.vsi.org).

All transferred deliverables, including those specified in the *VSIA Deliverables Document*, should be described in this section.

This deliverable is Mandatory (M).

## 2.5 Application Information

This section contains specific information about the use of the VC in real applications, including information helpful in the evaluation/selection step. This section is the repository for usage history and known bugs.

### 2.5.1 Version History

The VC provider must supply a version history of the VC describing changes and version releases associated with the VC. This information should clearly identify the current level of the release and be coordinated with the claims. This deliverable is Mandatory (M).

### 2.5.2 Known Bugs

A record of bugs, issues, and other problems known to exist with a particular VC must be maintained and made available to the VC user. This record should also include work-arounds and plans for fixes, if available. This deliverable is Mandatory (M).

### 2.5.3 Application Notes

Application notes are used to facilitate the VC user's adoption of the VC into his/her specific application environment. Design implementation options should be thoroughly explained. Examples should be given wherever appropriate. For example, a microprocessor unit (MPU) VC could offer multiple memory access mechanisms, such as burst access or pipeline access. The application information should thoroughly explain the tradeoffs between alternate memory subsystem implementations and should provide reference designs and evaluation aids as appropriate. Where core sets are provided for specific applications, details of their interrelations and interoperation should be fully explained and documented.

Application notes should include as many of the following sections as relevant and deemed pertinent to the VC being described.

#### 2.5.3.1 Frequently Asked Questions

This section includes the commonly asked questions about the VC and the relevant answers. This deliverable is Recommended (R).

#### 2.5.3.2 Design for Different Goals

This section includes descriptions of how to optimize the VC implementation for power, area, performance, and other goals. This deliverable is Recommended (R).

### 2.5.3.3 Related/Family VC

This section lists VCs that may be complementary to the VC being described with information showing how they may be utilized to better meet specific design goals. This deliverable is Recommended (R).

### 2.5.3.4 Customization Options

For VCs with customization options, this section provides a set of descriptions and instructions for using the different options to achieve different design objectives. This deliverable is Recommended (R).

### 2.5.3.5 Example Systems

Example systems are one of the best ways to demonstrate intended applications for the VC being described. Well-constructed examples can provide insight into the possible impact of the VC on overall system cost/performance. These examples also serve as references for configuring the system for the desired application. This deliverable is Recommended (R).

## 2.6 Test Information

This section contains information to quantify the successful integration of the VC into the target system chip. It includes data to support manufacturing test and related test activities in physical form. The manufacturing test deliverables are provided in Section 1.6 [reference 1]. These deliverables should be briefly discussed in this section.

The key information presented in this section should cover the testing approach for the VC. The methods of test techniques, such as full scan, partial scan, boundary scan, primary pins or other pins, need to be explained. The scope of the test coverage should be included with this key information.

### 2.6.1 Test Strategy

This section describes the socket interface for the VC and the related test methodology. The test strategy provides a high level overview of the recommended approach for achieving test coverage while minimizing cost factors.

The test strategy should be described using keywords such as: functional, scan (full or partial), built-in self test (BIST), or idd quiescent current (IDDQ). The VC provider may define additional keywords as necessary: test collar, core isolation, logic BIST, and so forth.

Section 1.6 [reference 1] contains appendices that should be consulted for a sample VC test strategy form showing the recommended table format.

This deliverable is Mandatory (M) for Hard VCs. Soft and Firm VCs are Conditional Mandatory (CM). If the Soft and Firm strategies are applicable and equivalent to a prior implementation, they become Mandatory (M).

### 2.6.2 Test Modules

The VC provider's deliverables to the test integrator shall include descriptions of the VC test methodology to the extent necessary for test integration and for silicon diagnostics.

A test module is an encapsulation of a test protocol. The test protocol specifies precisely how a test is to be performed. The test module contains additional information about intended usage, fault coverage, constraints on how the test may be used, diagnostic information associated with the test, and format of test language such as BSDL, STIL, IEEE1500 (P1500), etc.

The test module supplements the test strategy with information a test integrator will need for appropriate integration.

For further details of test modules, refer to Section 1.6 [reference 1].

This deliverable is Mandatory (M) for Hard VCs. Soft and Firm VCs are Conditional Mandatory (CM). If the Soft and Firm strategies are applicable and equivalent to a prior implementation, they become Mandatory (M).

### 2.6.3 Test Modes

Test modes describe an alternative mode of operation specifically designed for testing a VC. (The test mode is a different state than normal operation.)

Information about the test modes is essential to enable the test integrator to utilize the test mode and then to avoid unintended usage. The VC provider shall provide a description of each test mode.

This deliverable is Mandatory (M).

### 2.6.4 Mixed Signal Test Integration

This section describes mixed signal test integration for an analog VC.

#### 2.6.4.1 Signal Requirements for Analog Blocks

This section provides information about the digital vectors required to put the VC in test mode or to set up the VC to a particular state, preferably in Verilog change dump (VCD) format. These vectors can be supplied through a scan chain.

Digital signals that create analog signals in the VC should be supplied in a separate file. This file should contain a header specifying at least the word length and organization, most significant bit (MSB) or least significant bit (LSB) first. The sampling frequency, if not derived from the digital clock, should also be specified.

Wherever possible, test access should be listed, scan or otherwise, to provide the VC User with enough information to effectively test the analog portion of the VC.

Additional information should include:

- A description of how observability enables production of an analog fault signature during test
- Analog signals specified by a set of parameters that completely determine these signals (e.g., frequency, amplitude, phase, DC component)
- Special requirements for test signals (e.g., minimum bandwidth paths, load conditions)
- Accuracy requirements of measuring equipment necessary to measure analog signal
- Information describing conditions to set up test and how measurements are taken

This deliverable is Conditional Mandatory (CM) for Hard VCs. Hard VCs become Mandatory (M) only if they are analog mixed signal VCs. This deliverable is not applicable to Soft and Firm VCs.

#### 2.6.4.2 Analog Test Requirements

Tests must be supplied which provide a manufacturing test of functionality and validate conformance to specification. Published specifications, not guaranteed by the included test suite, must be labeled as not covered.

Any test that requires circuitry external to the VC should include the circuitry schematic. This circuitry should be limited to circuits that can be applied to external padded-out pins or be connected through standard analog scan type buses.

Support of IDDQ test of digital parts should be documented (e.g., digital outputs from analog blocks should be set at high or low level) and setup sequences and ranges of acceptable results for IDDQ in given processes should be clearly identified.

When possible, estimates of the test times needed for each test should be provided. Device set up and settling times should be separately identified. This will allow the test integrator to estimate the final manufacturing test time, or to skip tests which would be unacceptably long in production.

This deliverable is Conditional Mandatory (CM) for Hard VCs. Hard VCs become Mandatory (M) only if they are analog mixed signal VCs. This deliverable is no applicable to Soft and Firm VCs.

## **2.7 Supplier Information**

This section contains the VC provider's complete contact details, a statement of ownership, terms of use, limits of liability, and generic conformance and quality statements.

### **2.7.1 VC Provider Contact Information**

This section contains VC provider contact information, including address, email address, web URL, contact names for sales, support, licensing etc. This deliverable is Mandatory (M)

### **2.7.2 Transfer Package Information**

This section contains:

- Title, revision, and publication date of the deliverables
- Organization of the deliverables in terms of transfer media types and archive/file structure
- Part numbers and product nomenclature

This deliverable is Mandatory (M).

### **2.7.3 Standard Terms and Conditions**

This section contains standard terms and conditions governing the evaluation and subsequent incorporation of VC provider intellectual property into VC user systems including:

- Evaluation procedures
- Technical support and training
- Applicable patents, trademarks, and copyrights
- Business model
- Licensing requirements
- Applicable royalties
- Export limitations
- Warranties, liability limitations, disclaimers

This deliverable is Mandatory (M).

### **2.7.4 Third Party Reference**

This section contains complete contact information for third parties, testing labs and other partners who are willing to assist the VC user through provision of VC validation, VC integration, VC fabrication, and/or other services.

#### **2.7.4.1 Supporting VC Partner**

This section identifies partners (and their complete contact information) who are willing to supply one or more services such as: design service, EDA support, foundry service, or consulting service.

This deliverable is Recommended (R).

#### **2.7.4.2 ISV (Independent Software Vendor)**

This section lists third party firms (and their complete contact information) that supply software products for the VC such as: operating systems, software drivers, compilers, and application software.

This deliverable is Recommended (R).

# Appendix. Example of Structural Diagram

The following diagram is an example of a structural diagram. Since different blocks within the same diagram can span a hierarchy of detail, a hierarchy tagging number is recommended to make clear which level is represented by each block. A system-on-a-chip designed with mixed signal blocks could be tagged as shown in Figure 2: Sample Structural Diagram.

The amount of detail represented by each block varies with its level in the hierarchy. At higher levels, only the essential signals describing the function being considered are shown. At lower levels, control signals and data busses are shown. Interface bars may be added to make changes in hierarchy from one block to the next clearer. It is recommended that all inputs be shown on the left and outputs on the right.

**Figure 2: Sample Structural Diagram**

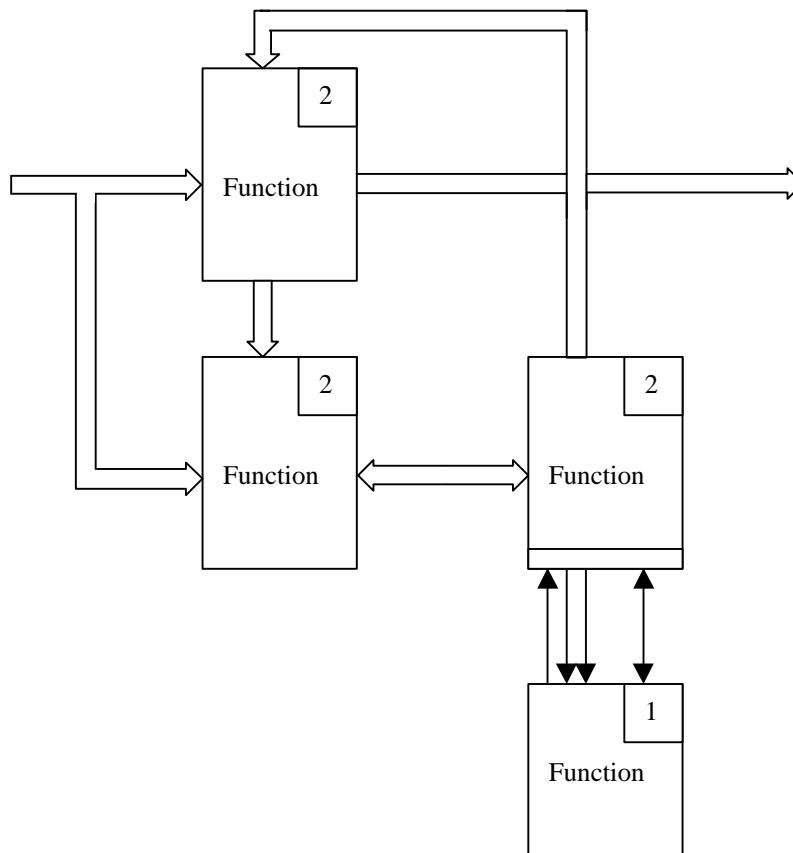


Table 2 shows a suggested hierarchy for the level tags (following the style of [reference 5] in Section 1.6).

**Table 2: Suggested Tagging Hierarchy**

7	VC function (no implementation information)	One large block
6	Some implementation information	Algorithm, firmware & software blocks
5	Full implementation information - complex blocks	MPEG, compressor, PLL, A/D
4	Composite units	Register, multiplier, flip-flop
3	Gate level	AND, NAND, etc.
2	Transistor level	NMOS, CMOS, R, C, etc.
1	Process level devices	Device elements, equations, etc.



