

# IP-XACT Components

v1.0

## Reference Manual



# IP-XACT Components

## Reference Manual

Copyright © 2007 ARM Limited. All rights reserved.

### Release Information

The following changes have been made to this book.

Change History			
Date	Issue	Confidentiality	Change
29 October 2007	A	Non-Confidential	First release

### Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

### Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

### Product Status

The information in this document is final, that is for a developed product.

### Web Address

<http://www.arm.com>

# Contents

## IP-XACT Components Reference Manual

<b>Chapter 1</b>	<b>Introduction</b>	
1.1	The IP-XACT standard .....	1-2
1.2	Benefits of using the IP-XACT standard .....	1-3
<b>Chapter 2</b>	<b>IP-XACT Content Specification</b>	
2.1	Component document .....	2-2
2.2	Bus definition document .....	2-6
2.3	Design document .....	2-8



# Chapter 1

## Introduction

This chapter provides a general description of the IP-XACT standard and the benefits of its use.

- *The IP-XACT standard* on page 1-2
- *Benefits of using the IP-XACT standard* on page 1-3.

## 1.1 The IP-XACT standard

IP-XACT is a standard for the description of electronic *Intellectual Property* (IP). This standard is intended for use by *Electronic Design Automation* (EDA) and *Electronic System Level* (ESL) tools. It was created and is owned by the SPIRIT Consortium. The primary goals of the standard are to:

- enable IP vendors to provide a single description of their components to all of their customers, regardless of the design language or tools that they use
- enable developers to transfer designs between environments that use different design languages.

IP-XACT includes an XML schema that defines a number of document types, and a set of semantic rules that describe the relationships between those documents. The most important document types are:

- design documents
- component documents
- bus definition documents.

## 1.2 Benefits of using the IP-XACT standard

The electronics industry uses tools produced by a number of vendors in the design process. Many of these tools use unique and proprietary formats. A standard for description of electronic design information lets developers exchange this information quickly between different design environments. This speeds the design flow and leads to quicker implementation.

The IP-XACT standard provides similar benefits to the debug process. Simple systems can use a single interface with direct access to the processor for debug purposes. Engineers can easily describe such systems to debuggers. However, complex systems can contain many processors and buses, and even their own embedded debug subsystems. Because most debuggers, like design tools, use proprietary formats for design information, correctly describing a complex system to a debugger can be very difficult. A standardized description of a system can save time and minimize errors in the debug process, even if you use debuggers that cannot directly process an IP-XACT description of a system.

ARM provides IP-XACT descriptions of its components to facilitate importing those components into EDA tools and debuggers that support IP-XACT. Broader adoption of the IP-XACT standard throughout the electronics industry permits ARM components to be more easily integrated into designs that contain components from other vendors.





## Chapter 2

# IP-XACT Content Specification

IP-XACT is a standard that specifies how to describe different types of electronic IP in the form of an XML document. This chapter describes the documents used to describe ARM products.

This chapter contains the following sections:

- *Component document* on page 2-2
- *Bus definition document* on page 2-6
- *Design document* on page 2-8.

## 2.1 Component document

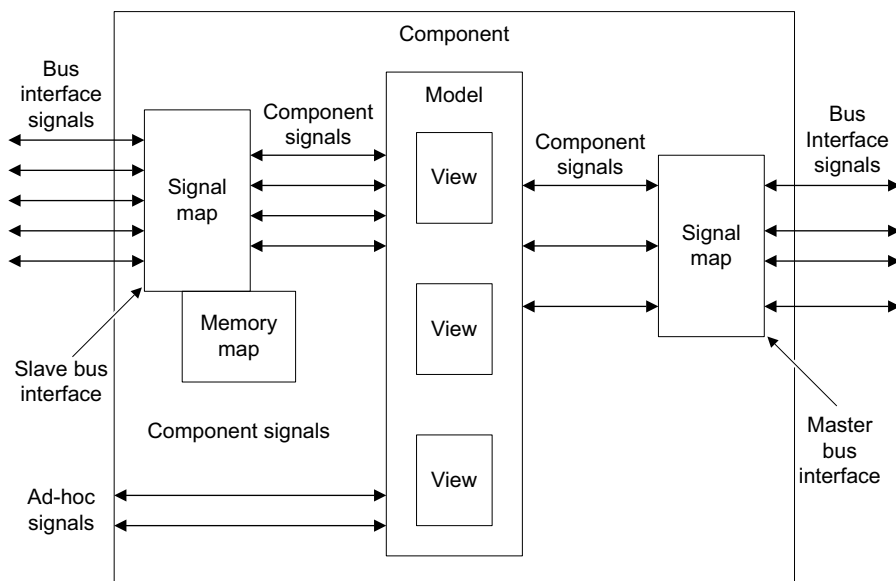
A component document represents a single IP block that you can instantiate as a single entity in a design. Components have bus interfaces whose types are specified by a bus definition. A component document can also define one or more views of the implementation, or an interconnect infrastructure in the form of a bridge or a channel.

Typically the component document contains the following elements:

- A model element represents the top-level signal list for that component.
- Signal map elements map signals to appropriate bus interfaces.
- Other elements document the memory map for slave bus interfaces and provide information on accessible locations and their structure. These locations are called *registers* in the IP-XACT standard. You can implement these as functional or *Register Transfer Level (RTL)* registers.

Components can also contain a hierarchical collection of IP blocks as an implementation of the component. However, this is not well supported by all EDA tools.

Figure 2-1 shows a simplified view of an IP-XACT component, together with its major features.



**Figure 2-1 Structure of a component in IP-XACT**

This section also describes:

- *Component description common elements*
- *Component description conditional elements* on page 2-4
- *Component description unused features* on page 2-5.

### 2.1.1 Component description common elements

Table 2-1 lists the common elements for each component description.

**Table 2-1 Common elements for component description**

Feature	Element	Description
A <i>Vendor, Library, Name, and Version</i> (VLNV) appropriate to the component.	spirit:vendor spirit:library spirit:name spirit:version	These elements reference the IP-XACT description. These are the only required elements from the IP-XACT standard.
A bus interface for each major grouping of signals exported from the top-level RTL.	spirit:busInterface	A bus interface is based on standard interfaces if possible.
A signal map linking bus interface signals with the model signals.	spirit:signalMap	Each bus interface has a signal map. This element permits tools to perform top-level stitching.
A model with at least one view representing the RTL for the component. The spirit:modelName element for this view contains the name of the top-level module from the RTL.	spirit:model spirit:modelName spirit:modelParameter spirit:view	spirit:model is required because it contains the RTL signals list. The RTL view references an RTL fileset. This is important for RTL stitching tools.
All signals as exported from the top-level RTL.	spirit:signal	These signals match the RTL.
Each fileset comprising the component. The fileset is referenced by the RTL view.	spirit:fileSet	<p>The path to the fileset is relative to an environment variable, for example:            \${MY_LOCATION}/path/file.v</p> <p>This element takes automated compilation of designs into account without the requirement to locate RTL files manually.</p> <p><b>Note</b></p> <p>Filesets are confidential and are not released in the public domain.</p>

Table 2-1 Common elements for component description (continued)

Feature	Element	Description
Each referenced RTL file.	spirit:file	The path to the file is relative to an environment variable. The variable is completed by the <i>Design Environment</i> (DE).
Parameters are given only where they appear in the RTL, provided they do not add structural features to the design, such as additional components or signals.	spirit:parameter	IP-XACT files represent a realized block with all parameters resolved. However, if a component has multiple configurations, parameters can be useful to avoid having separate IP-XACT files for each parameter variant.
Memory maps, address blocks, and memory-mapped registers.	spirit:memoryMap spirit:addressBlock spirit:register	These elements give information about the programmer's view of the component. Only memory-mapped locations are included. Processor registers that are not mapped to memory are specifically excluded.
Channels or bridges that represent a fabric channel.	spirit:channel spirit:bridge	This information describes the internal mapping between bus interfaces. This permits higher level tools to create the memory map of a complete system with much less intervention.

2.1.2 Component description conditional elements

Some elements are included in a component document if a condition is satisfied. Table 2-2 lists these elements and the condition that causes them to be included.

Table 2-2 Conditional elements for component description

Condition	Element	Description
The component has one or more bus master interfaces.	spirit:addressSpace	This defines a logical space that is referenced by a bus master. It includes information about the size of the space and any executable images to be loaded.
The component represents a processor or has one or more cores.	spirit:cpu	This identifies the component as a processor. It specifies the core instance name and enables tools to identify the component type more easily.

### 2.1.3 Component description unused features

Certain features available in IP-XACT are never used in ARM IP-XACT component descriptions. Table 2-3 lists these.

**Table 2-3 Excluded features for component description**

Feature	Description
Timing and other signal constraints	These are process and technology-independent timing constraints used in synthesis or basic timing budgeting. ARM considers that the current constraints are not process-neutral and that the <i>Synopsys Design Constraints</i> (SDC) file format must be used as an alternative.
Clock drivers	Drivers permit a clock pulse to be described. They are a reference for timing constraints.
Vendor extensions	These are any additional information related to the design that is not part of the standard today.
Verification components	These components enable a verification environment to be automatically constructed around a block. The IP-XACT standard is not mature in this area. Existing tools can make use of this information but only if presented in a way that the particular tool recognizes.
Software	This includes driver code, boot code, or other software with the component so that it can be automatically assembled.
Hierarchical design components	These provide pre-connected groups of components. This feature is not used because of limited tool support.

2.2 Bus definition document

The bus definition document describes a bus type. It describes the signals in the bus interface and the constraints that apply to those signals. This includes signal names, direction, width, and usage. Bus definitions can also describe constraints on the use of that bus type, such as how many bus masters are allowed on each bus.

Many standard definitions exist for well-known buses. ARM provides IP-XACT bus definitions for AMBA on <http://www.amba.com>. ARM strongly recommends, for inter-operability reasons, that you use these if you are developing IP-XACT descriptions with AMBA interfaces.

An IP-XACT bus definition represents signals that implement a single protocol. The definition describes all the signals that participate in that protocol. However, some component signals such as clock signals can participate in more than one bus interface and appear in multiple bus definition documents.

A component requires new bus definitions if it has proprietary connections with other components.

This section also describes:

- *Bus definition document common elements*
- *Bus definition document unused features* on page 2-7.

2.2.1 Bus definition document common elements

Table 2-4 lists the common elements for each bus definition document.

Table 2-4 Common elements for bus description

Feature	Element	Description
A VLNV appropriate to the component.	spirit:vendor spirit:library spirit:name spirit:version	These are the only required elements from the IP-XACT standard.
Signals	spirit:signal	This is required for any signal-level bus.
Signal width and direction	spirit:width spirit:direction	These specify the direction and the width for all signals, including single wires. An exception is for signal wires that have an unconstrained width in the specification.
Signal markers	spirit:isClock spirit:isData spirit:isAddress	These elements identify the type of information carried by the signal wire.

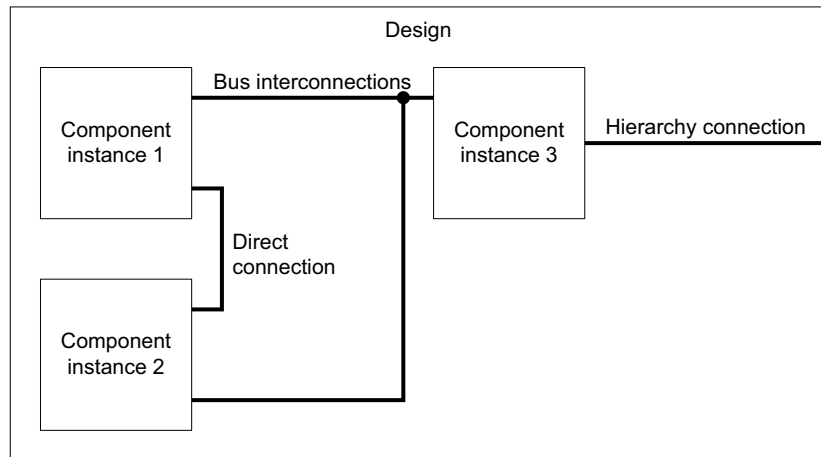
### **2.2.2 Bus definition document unused features**

ARM bus definition documents use all IP-XACT supported features except for vendor extensions.

## 2.3 Design document

The design document represents instances of components in a system and the interconnection between these instances. Figure 2-2 shows an example system with three component instances connected through buses.

A design document acts as a layer for instantiating components and connecting them together. Although you can reference designs from a view in a higher level component to create designs with an arbitrarily deep hierarchy, this is not presently well supported by some tools.



**Figure 2-2 Design representation in IP-XACT**

This section also describes:

- *Design document common elements* on page 2-9
- *Design document conditional elements* on page 2-9
- *Design document unused features* on page 2-9.



### 2.3.1 Design document common elements

Table 2-5 lists common elements in a design document.

**Table 2-5 Common elements for design description**

Feature	Element(s)	Description
A VLVN appropriate to the component.	spirit:vendor spirit:library spirit:name spirit:version	These are the only required elements from the IP-XACT standard.
Instances	spirit:componentInstance	This element is required to permit any kind of design to be created.
Interconnect	spirit:interconnection	The bus interface interconnection between instances is required to allow automated RTL stitching.

### 2.3.2 Design document conditional elements

Some features are included in a design document if a condition is satisfied. Table 2-6 lists these elements and the condition that causes them to be included.

**Table 2-6 Conditional elements for design description**

Feature	Element(s)	Description
Hierarchy connections, that is, external bus connections.	spirit:hierConnection	This enables the design to export buses to its containing design.
Ad-hoc connections, that is, signals that are not connected through bus interfaces.	spirit:adHocConnection	This enables signals that are not part of a bus to be connected between blocks.

### 2.3.3 Design document unused features

ARM design documents use all IP-XACT supported features except for vendor extensions.

