Verification Continuum™

VC Verification IP
AMBA AXI
UVM Getting Started Guide

Version S-2021.06, June 2021

SYNOPSYS®

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at http://www.synopsys.com/company/legal/trademarks-brands.html.
All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

www.synopsys.com

Contents

Preface

About This Document

This Getting Started Guide presents information about integrating the VC VIP for AXI (referred to as VIP) into testbenches that are compliant with the SystemVerilog Universal Verification Methodology (UVM). You are assumed to be familiar with the AMBA AXI protocol and UVM.

Web Resources

- Documentation through SolvNet: https://solvnetplus.synopsys.com (Synopsys password required)
- Synopsys Common Licensing (SCL): http://www.synopsys.com/keys

Customer Support

To obtain support for your product, choose one of the following:

1. Go to https://solvnetplus.synopsys.com and open a case.

   Enter the information according to your environment and your issue.
2. Send an e-mail message to support_center@synopsys.com.

   Include the Product name, Sub Product name, and Tool Version in your e-mail so it can be routed correctly.
3. Telephone your local support center.
   - North America:

     Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.
   - All other countries:

     https://www.synopsys.com/support/global-support-centers.html

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

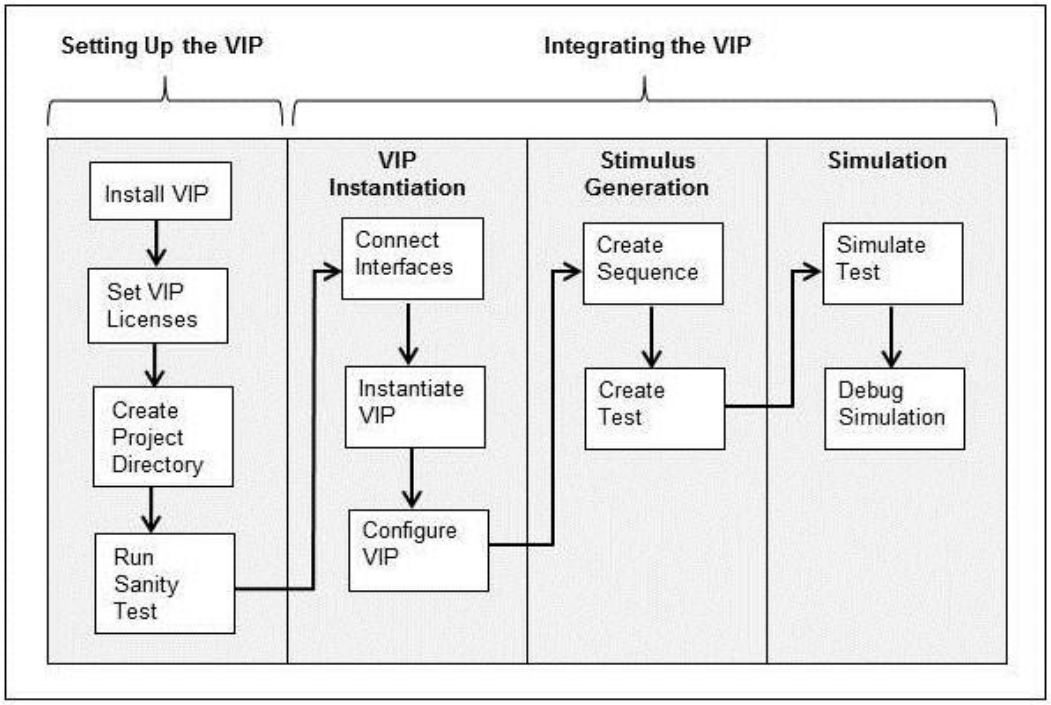
1 Overview of the Getting Started Guide

This Getting Started Guide presents information about integrating the VC VIP for AXI (referred to as VIP) into testbenches that are compliant with the SystemVerilog Universal Verification Methodology (UVM). Figure 1-1 is the VIP integration and test work flow presented in this document. The steps for setting up the VIP are documented in the *VC Verification IP Installation and Setup Guide*. This guide is available on the SolvNet Download Center (click here -> VC VIP Library -> R-2020.12-> Installation Guide) and in the VIP installation at the following location:

```
$DESIGNWARE_HOME/vip/svt/common/latest/doc/uvm_install.pdf
```

The VIP setup should be completed before executing the steps in this document.

Figure 1-1 VIP Integration and Test Work Flow



You are assumed to be familiar with the AMBA AXI protocol and UVM. For more information on the VIP, refer to the *VC Verification IP AMBA AXI UVM User Guide* on SolvNet (click here) or in the VIP installation at the following location:

```
$DESIGNWARE_HOME/vip/svt/vip_title/latest/doc/axi_svt_uvm_user_guide.pdf
```

2 Integrating the VIP into a User Testbench

The VC VIP for AXI provides a suite of advanced SystemVerilog verification components and data objects that are compliant to UVM. Integrating these components and objects into any UVM compliant testbench is straightforward. For a complete list of VIP components and data objects, refer to the main page of the *VC VIP AXI Class Reference* (only in HTML format) at the following location:

```
$DESIGNWARE_HOME/vip/svt/vip_title/latest/doc/axi_svt_uvm_class_reference/html/
index.html
```

2.1 VIP Testbench Integration Flow

The AXI system environment (svt_axi_system_env) is the top-level component provided by the VIP. This environment encapsulates all of the VIP components and implicitly constructs the required number of AXI master and AXI slave agents as specified by its system configuration object. You can instantiate and construct the AXI system environment in the top-level environment of your UVM testbench.

Figure 2-1 Top-level Architecture of an AXI VIP Testbench

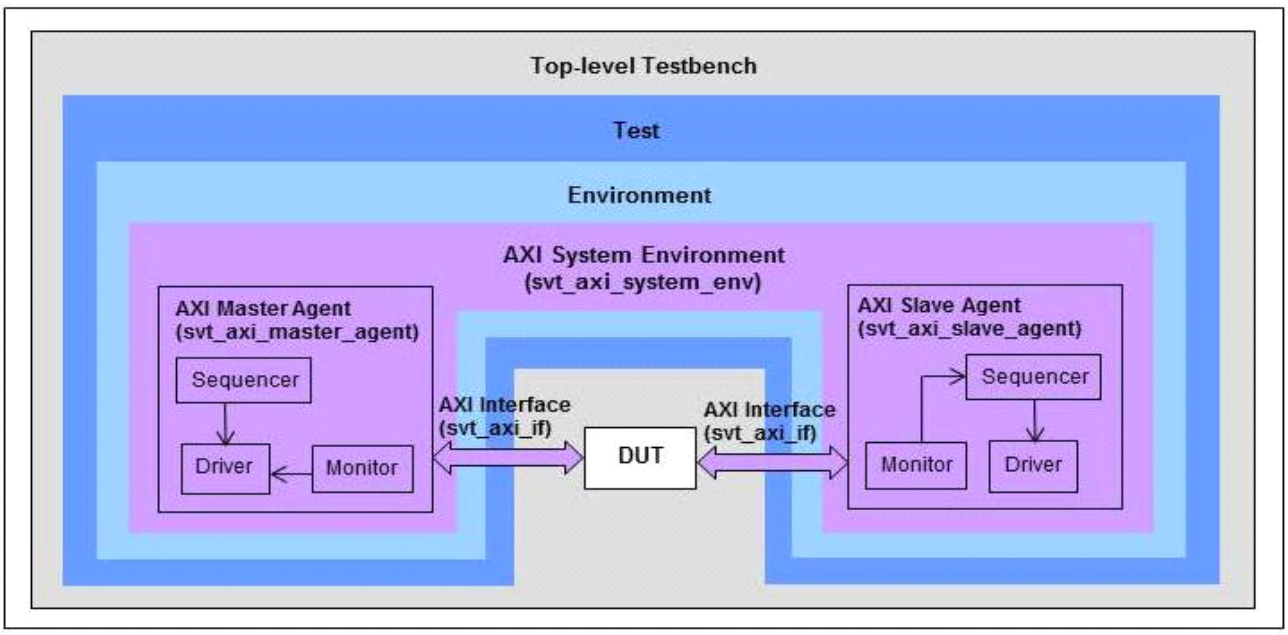


Figure 2-1 is a top-level architecture of a simple VC VIP for AXI testbench. The steps for integrating the VIP into a UVM testbench are described in the following sections:

- "Connecting the VIP to the DUT"
- "Instantiating and Configuring the VIP"
- "Creating a Test Sequence"
- "Creating a Test"

The code snippets presented in this chapter are generic and can be applied to any UVM compliant testbench. For more information on the code usage, refer to the following example:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog/
tb_axi_svt_uvm_basic_sys
```

2.1.1 Connecting the VIP to the DUT

The following are the steps to establish a connection between the VIP to the DUT in your top-level testbench:

- Include the standard UVM and VIP files and packages.

```
`include "uvm_pkg.sv"
`include "svt_axi.uvm.pkg"
`include "svt_axi_if.svi" //top-level AXI interface

import uvm_pkg::*;
import svt_uvm_pkg::*;
import svt_axi_uvm_pkg::*;
```

- Instantiate and connect the top-level AXI interface to a system clock.

```
svt_axi_if axi_if();
assign axi_if.common_aclk = <system clock>;
```

- Connect the top-level AXI interface to the DUT and the AXI system environment.

```
dut dut_inst(axi_if);
uvm_config_db#(svt_axi_vif)::set(uvm_root::get(),
"uvm_test_top.env.axi_system_env", "vif", axi_if);
```

The `uvm_config_db` command connects the top-level AXI interface to the virtual interface of the AXI system environment. The "`uvm_test_top`" represents the top-level module in the UVM environment. The "`env`" is an instance of your testbench environment. The "`axi_system_env`" is an instance of the AXI system environment (svt_axi_system_env).

- Connect the reset pin of the AXI master and slave agent interfaces to the desired reset signals.

```
assign axi_if.master_if[0].aresetn = <User-select Reset
Signal>;
assign axi_if.slave_if[0].aresetn = <User-select Reset
Signal>;
```

2.1.2 Instantiating and Configuring the VIP

The following are steps to instantiate and configure the AXI system environment in your testbench environment.

- Instantiate the AXI system environment (`svt_axi_system_env`) in the build phase of your testbench environment.

```
svt_axi_system_env axi_system_env;
axi_system_env =
svt_axi_system_env::type_id::create("axi_system_env", this);
```

- Create a customized AXI system configuration class by extending the AXI system configuration class (`svt_axi_system_configuration`) and specifying the required configuration parameters.

  For example:

```
class cust_svt_axi_system_configuration extends
svt_axi_system_configuration;
  function new (string name =
"cust_svt_axi_system_configuration");
    super.new(name);

    // Create a single AXI master agent and a single slave agent
    this.num_masters = 1;
    this.num_slaves  = 1;

    // Create port configurations
    this.create_sub_cfgs(1,1);

    this.master_cfg[0].data_width = 256;
    this.slave_cfg[0].data_width = 256;
    this.master_cfg[0].id_width = 4;
    this.slave_cfg[0].id_width = 4;
  endfunction
endclass
```

For more information on the configuration class, refer to the `svt_axi_system_configuration` and `svt_axi_port_configuration` *Class References* at the following locations:

```
$DESIGNWARE_HOME/vip/svt/vip_title/latest/doc/
axi_svt_uvm_class_reference/html/
class_svt_axi_system_configuration.html

$DESIGNWARE_HOME/vip/svt/vip_title/latest/doc/
axi_svt_uvm_class_reference/html/
class_svt_axi_port_configuration.html
```

- Construct the customized AXI system configuration and pass the configuration to the AXI system environment (instance of `svt_axi_system_env`) in the build phase of your testbench environment.

```
cfg =
cust_svt_axi_system_configuration::type_id::create("cfg");
uvm_config_db#(svt_axi_system_configuration)::set(this,
"axi_system_env", "cfg", cfg);
```

The "`cust_svt_axi_system_configuration`" is the customized AXI system configuration as defined in the previous step. The "`cfg`" is an instance of this configuration.

2.1.3 Creating a Test Sequence

The VIP provides a base sequence class for the AXI master agent (svt_axi_master_base_sequence) and the AXI slave agent (svt_axi_slave_base_sequence). You can extend these base sequences to create test sequences for the AXI master and slave agents.

For more information on the AXI master and slave base sequences, and the VIP sequence collection, refer to the Sequence Page of the *VC VIP AXI Class Reference* at the following location:

```
$DESIGNWARE_HOME/vip/svt/vip_title/latest/doc/
axi_svt_uvm_class_reference/html/sequencepages.html
```

In addition, a list of random and directed sequences are available in the VIP examples. For more information on the example sequences, refer to the example directories at the following location:

```
$DESIGNWARE_HOME/vip/svt/vip_title/latest/examples/sverilog
```

Note

You must set a slave response sequence for active slaves in the run phase.

2.1.4 Creating a Test

You can create a VIP test by extending the uvm_test class. In the build phase of the extended class, you construct the testbench environment and set the respective AXI master and slave sequences.

```
class random_wr_rd_test extends uvm_test;

  // build_phase
  env = axi_basic_env::type_id::create("env", this);

  uvm_config_db#(uvm_object_wrapper)::set(this,
  "env.axi_system_env.master*.sequencer.main_phase",
  "default_sequence",
axi_master_wr_rd_sequence::type_id::get());

  uvm_config_db#(uvm_object_wrapper)::set(this,
  "env.axi_system_env.slave*.sequencer.run_phase",
  "default_sequence",
  axi_slave_mem_response_sequence::type_id::get());
```

2.2 Compiling and Simulating a Test with the VIP

The steps for compiling and simulating a test with the VIP are described in the following sections:

- “Directory Paths for VIP Compilation”
- “VIP Compile-time Options”
- “VIP Runtime Option”

2.2.1 Directory Paths for VIP Compilation

You need to specify the following directory paths in the compilation commands for the compiler to load the VIP files.

```
+incdir+project_directory_path/include/sverilog
+incdir+project_directory_path/src/sverilog/simulator
```

Where, *project_directory_path* is your project directory and *simulator* is vcs, ncv or mti.

For example:

```
+incdir+/home/project1/testbench/vip/include/sverilog
+incdir+/home/project1/testbench/vip/src/sverilog/vcs
```

2.2.2 VIP Compile-time Options

The following are the required compile-time options for compiling a testbench with the VC VIP for AXI:

```
+define+SVT_UVM_TECHNOLOGY
+define+UVM_PACKER_MAX_BYTES=1500000
+define+UVM_DISABLE_AUTO_ITEM_RECORDING
+define+SYNOPSYS_SV
```

Note

`UVM_PACKER_MAX_BYTES define` needs to be set to maximum value as required by each VIP title in your testbench. For example, if VIP title 1 needs `UVM_PACKER_MAX_BYTES` to be set to 8192, and VIP title 2 needs `UVM_PACKER_MAX_BYTES` to be set to 500000, you need to set `UVM_PACKER_MAX_BYTES` to 500000.

Macro	Description
SVT_UVM_TECHNOLOGY	Specifies SystemVerilog based VIPs that are compliant with UVM
UVM_PACKER_MAX_BYTES	Sets to 1500000 or greater
UVM_DISABLE_AUTO_ITEM_RECORDING	Disables the UVM automatic transaction begin and end event triggering and recording Refer to Section 6.9, “Why the User Needs to Disable Auto Item Recording“ of the *VC Verification IP AMBA AXI UVM User Guide* for more information
SYNOPSYS_SV	Specifies SystemVerilog based VIPs that are compliant with UVM

The following compile-time option is required if and only if you have created a user-defined file to override the VIP default maximum values of the system constants such as maximum delay values and maximum address width.

```
+define+SVT_AXI_INCLUDE_USER_DEFINES
```

For more information, refer to Section 3.3.7, "Overriding System Constants" of the *VC Verification IP AMBA AXI UVM User Guide*.

2.2.3 VIP Runtime Option

No VIP specific runtime option is required to run simulations with the VIP. Only relevant UVM runtime options are required.

For example:

```
+UVM_TESTNAME=random_wr_rd_test
```

A Summary of Commands, Documents, and Examples

A.1 Commands in This Document

Display VIP models and examples under the VIP installation directory specified by $DESIGNWARE_HOME:

```
% $DESIGNWARE_HOME/bin/dw_vip_setup -info home
```

Add VIP models to the project directory:

```
% $DESIGNWARE_HOME/bin/dw_vip_setup  -path project_directory -add
VIP_model -svlog
```

Add VIP examples to the directory where the command is executed:

```
% $DESIGNWARE_HOME/bin/dw_vip_setup -e VIP_example -svlog
```

A.2 Primary Documentation for VC VIP AXI

VC Verification IP UVM Installation and Setup Guide:

```
$DESIGNWARE_HOME/vip/svt/common/latest/doc/uvm_install.pdf
```

VC VIP AXI UVM User Guide:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/axi_svt_uvm_user_guide.pdf
```

VC VIP AXI Getting Started Guide:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/axi_svt_uvm_getting_started.pdf
```

VC VIP AXI QuickStart:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog/
tb_axi_svt_uvm_basic_sys/doc/tb_axi_svt_uvm_basic_sys/index_basic.html
```

VC VIP AXI Class Reference:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/axi_svt_uvm_class_reference/html/
index.html
```

VC VIP AXI Verification Plans:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/VerificationPlans
```

A.3 Example Home Directory

Directory that contains a list of VIP example directories:

```
$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog
```

View simulation options for each example:

```
gmake help
```