Verification Continuum™

VC Verification IP
AMBA CHI
UVM Getting Started Guide

Version S-2021.06, June 2021

SYNOPSYS®

Contents

Preface

About This Document

This Getting Started Guide presents information about integrating the VC VIP for CHI (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 CHI 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

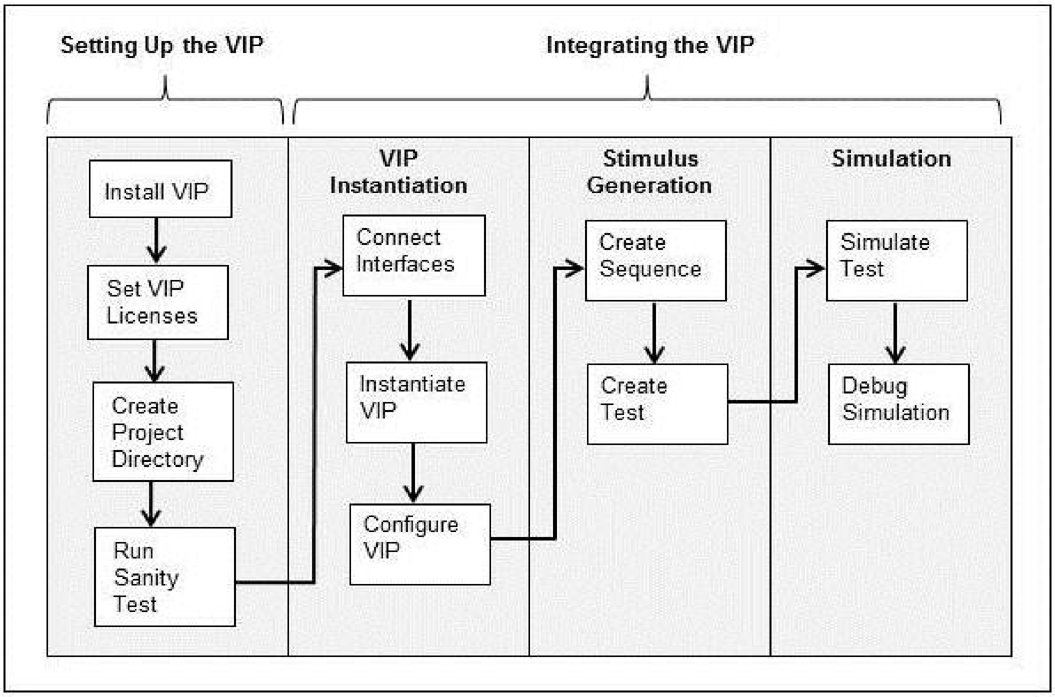
1 Overview of the Getting Started Guide

This Getting Started Guide presents information about integrating the VC VIP for CHI (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 UVM Installation and Setup Guide*. This guide is available on the SolvNet Download page (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 CHI protocol and UVM. For more information on the VIP, see the *VC Verification IP AMBA CHI UVM User Guide* on SolvNet (click here) or in the VIP installation at the following location:

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

2 Integrating the VIP into a User Testbench

The VC VIP for CHI 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, see the main page of the *VC VIP CHI Class Reference* (only in HTML format) at the following location:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_class_reference/html/index.html

2.1 VIP Testbench Integration Flow

The CHI system environment (`svt_chi_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 CHI Request Node (RN) and CHI Slave Node (SN) agents as specified by its system configuration object. You can instantiate and construct the CHI system environment in the top-level environment of your UVM testbench.

Figure 2-1 Top-level Architecture of a CHI VIP Testbench

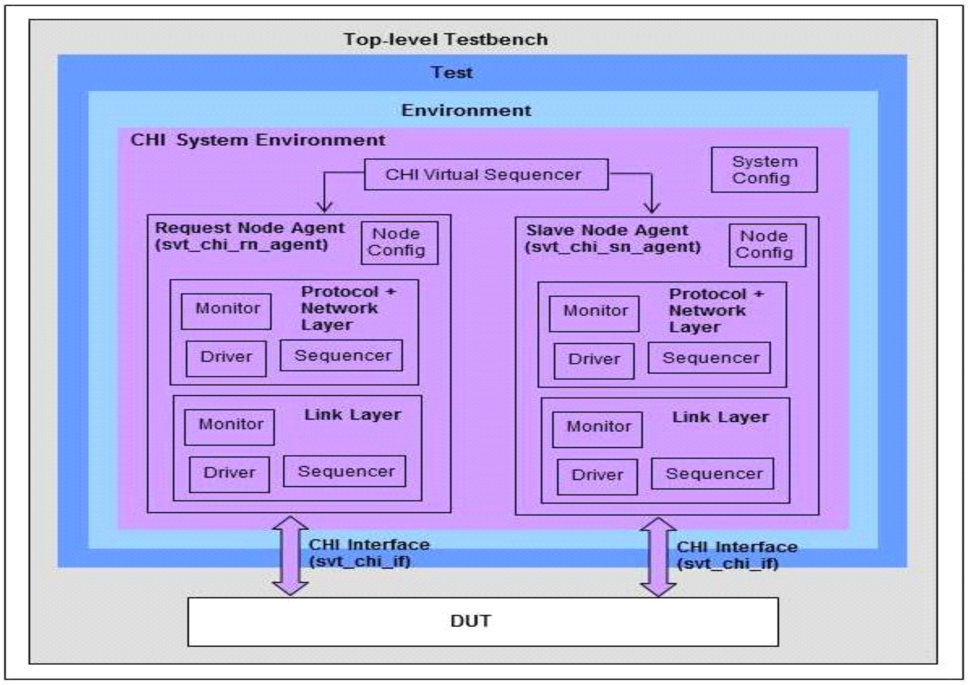


Figure 2-1 is a top-level architecture of a simple VC VIP for CHI testbench. The steps for integrating the VIP into a UVM testbench are described in the following sections:

- "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, see the following example:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog/tb_chi_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_chi.uvm.pkg"
`include"svt_chi_if.svi" //top-level CHI interface

import uvm_pkg::*;
import svt_uvm_pkg::*;
import svt_chi_uvm_pkg::*;
```

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

```
svt_chi_if chi_if(SystemClock, SystemReset);
```

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

```
dut dut_inst(chi_if);

uvm_config_db#(svt_chi_vif)::set(uvm_root::get(),
"uvm_test_top.env.chi_system_env", "vif", chi_if);
```

  The `uvm_config_db` command connects the top-level CHI interface to the virtual interface of the CHI 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 `chi_system_env` is an instance of the CHI system environment (`svt_chi_system_env`).

2.1.2 Instantiating and Configuring the VIP

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

- Instantiate the CHI system environment (svt_chi_system_env) in the build phase of your testbench environment.

```
svt_chi_system_env chi_system_env;
chi_system_env = svt_chi_system_env::type_id::create("chi_system_env",
this);
```

❖ Create a customized CHI system configuration class by extending the CHI system configuration class (`svt_chi_system_configuration`) and specifying the required configuration parameters.

For example,

```
class cust_svt_chi_system_configuration extends
svt_chi_system_configuration;
  function new (string name = "cust_svt_chi_system_configuration");
    super.new(name);

     //Allocate the RN and SN node configurations before a user sets
   //the parameters. This function is to be called if (and before)
   //the user sets the configuration parameters by setting each
   //parameter individually and not by randomizing the system
   //configuration.
   //Prototype of the method is:
   //void create_sub_cfgs(int num_chi_rn = 1,int num_chi_sn = 1,
   //int num_chi_ic_rn = 0,int num_chi_ic_sn = 0,
   //int num_axi_masters = 1,int num_axi_slaves = 1,
   //int num_axi_ic_master_ports = 0,
   //int num_axi_ic_slave_ports = 0);
    this.create_sub_cfgs(1,1,0,0,0,0,0,0);
//Configure the number of CHI Home Nodes in the CHI System ENV.
    this.num_hn = 8;

      //Configure the number of CHI Request Nodes in the CHI System
   //ENV.
    this.num_rn = 1;

   //Configure the number of CHI Slave Nodes in the CHI System ENV.
    this.num_sn = 1;

   //Set the interface type.
    rn_cfg[0].chi_interface_type = svt_chi_node_configuration::RN_F;
    sn_cfg[0].chi_interface_type = svt_chi_node_configuration::SN_F;

   //Set unique node id for each node
    rn_cfg[0].node_id = 0;
    sn_cfg[0].node_id = 6;

      //Set the width of Data field within Data VC Flit.
    rn_cfg[0].flit_data_width = 128;
    sn_cfg[0].flit_data_width = 128;
  endfunction
endclass
```

For more information on the configuration class, see the `svt_chi_system_configuration` and `svt_chi_port_configuration` *Class References* at the following locations:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_class_reference/html/class_svt_chi_system_configuration.html

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_class_reference/html/class_svt_chi_port_configuration.html

For more information on how to configure the host node (HN) address map and `node_id`, see the configuration file of the `tb_chi_svt_uvm_basic_sys` example at the following location:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog/tb_chi_svt_uvm_basic_sys/env/cust_svt_amba_system_configuration.sv

❖ Construct the customized CHI system configuration and pass the configuration to the CHI system environment (instance of `svt_chi_system_env`) in the build phase of your testbench environment.

```
        cfg = cust_svt_chi_system_configuration::type_id::create("cfg");

        uvm_config_db#(svt_chi_system_configuration)::set(this,
        "chi_system_env", "cfg", cfg);
```

The `cust_svt_chi_system_configuration` is the customized CHI 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 CHI RN agent (`svt_chi_rn_transaction_base_sequence`) and the CHI SN agent (`svt_chi_sn_transaction_base_sequence`). You can extend these base sequences to create test sequences for the CHI RN and SN agents.

For more information on the CHI RN and SN base sequences, and the VIP sequence collection, see the sequence page of the *VC VIP CHI Class Reference* at the following location:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_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, see the example directories at the following location:

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

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 CHI RN, SN and Snoop response sequences.

```
class random_wr_rd_test extends uvm_test;

  //Build phase
  virtual function void build_phase(uvm_phase phase);

    `uvm_info("build_phase", "is entered",UVM_LOW)

    super.build_phase(phase);

    env = chi_basic_env::type_id::create("env", this);

   //RN sequence
    uvm_config_db#(uvm_object_wrapper)::set(this,
       "env.chi_system_env.rn[0].rn_xact_seqr.run_phase",
       "default_sequence",
       svt_chi_rn_transaction_random_sequence::type_id::get());

   //SN sequence
    uvm_config_db#(uvm_object_wrapper)::set(this,
       "env.chi_system_env.sn[0].sn_xact_seqr.run_phase",
       "default_sequence",
       chi_sn_directed_response_sequence::type_id::get());

    //Snoop response sequence for RNF
    uvm_config_db#(uvm_object_wrapper)::set(this,
       "env.chi_system_env.sn[0].rn_snp_xact_seqr.run_phase",
       "default_sequence",
       chi_rn_directed_snoop_response_sequence::type_id::get());

  endfunction : build_phase
endclass
```

The `svt_chi_rn_transaction_random_sequence` is the RN sequence provided by the VIP. For more information on this sequence, see the following sequence page of the *VC VIP CHI Class Reference* at the following location:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_class_reference/html/ class_svt_chi_rn_transaction_random_sequence.html

The `chi_sn_directed_response_sequence` is a user-defined SN response sequence. For more information on creating SN response sequence, see the following file in the VIP example:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog/tb_chi_svt_uvm_basic_sys/env/ chi_sn_directed_response_sequence.sv

The `chi_rn_directed_snoop_response_sequence` is a user-defined snoop response sequence for RN.

For more information on creating Snoop response sequence, see the following file in the VIP example:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/examples/sverilog/tb_chi_svt_uvm_intermediate_sys/env/ chi_rn_directed_snoop_response_sequence.sv

Note
You must set a Slave Node (SN) response sequence for active slave nodes in the run phase. You must set a Snoop response sequence for each active Request Node (RN) in the run phase.

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 AMBA CHI:

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

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
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_CHI_INCLUDE_USER_DEFINES
```

For more information, see *Overriding System Constants* section of the *VC Verification IP AMBA CHI 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 CHI

VC VIP UVM Installation and Setup Guide:

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

VC VIP CHI UVM User Guide:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_user_guide.pdf

VC VIP CHI Getting Started Guide:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_getting_started.pdf

VC VIP CHI Class Reference:

$DESIGNWARE_HOME/vip/svt/amba_svt/latest/doc/chi_svt_uvm_class_reference/html/index.html

VC VIP CHI 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
```