Verification Continuum™

VC Verification IP
SWD
UVM Getting Started Guide

Version Q-2020.06, June 2020

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 https://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 SWD (referred to as VIP) into testbenches that are compliant with the SystemVerilog Universal Verification Methodology (UVM). You are assumed to be familiar with the SWD protocol and UVM.

Web Resources

- Documentation through SolvNetPlus: 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:

- Go to https://solvnetplus.synopsys.com and open a case.
  - Enter the information according to your environment and your issue.
  - For simulation issues, provide a UVM_FULL verbosity log file of the VIP instance and a VPD or FSDB dump file of the VIP interface.
- Send an e-mail message to support_center@synopsys.com
  - Include the Product name, Sub Product name, and Product version for which you want to register the problem.
- 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:

    http://www.synopsys.com/Support/GlobalSupportCenters

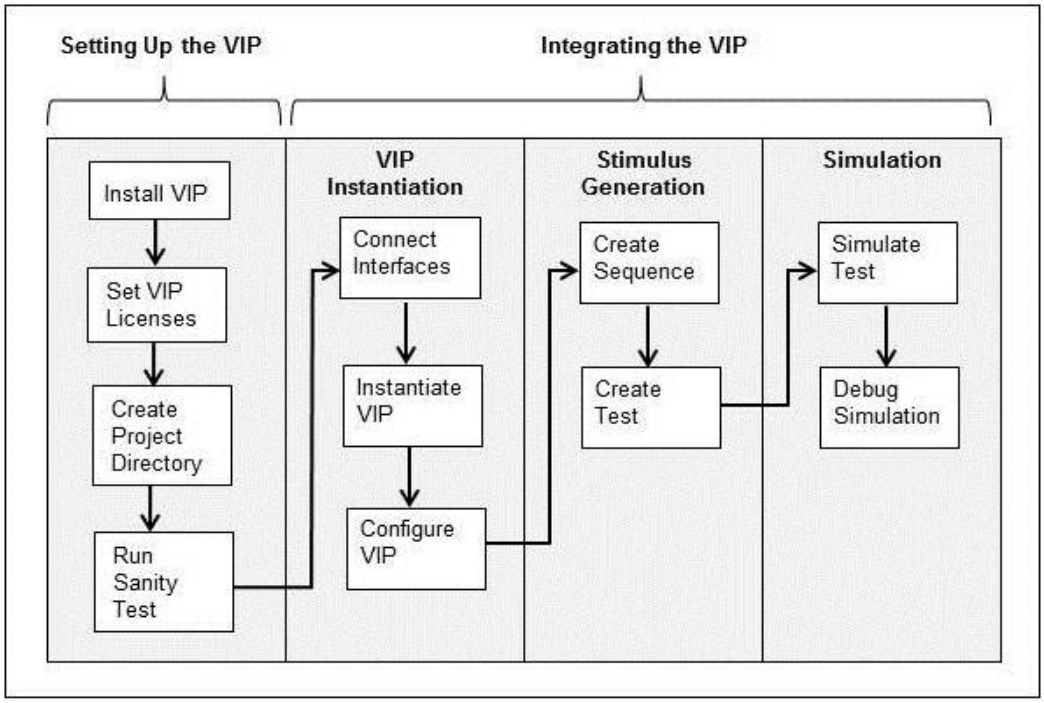
1 Overview of the Getting Started Guide

This Getting Started Guide presents information about integrating the VC VIP for SWD (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 Center (click here -> VC VIP Library -> Q-2020.03 -> 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 SWD protocol and UVM. For more information on the VIP, refer to the *VC Verification IP SWD UVM User Guide* on SolvNet (click here) or in the VIP installation at the following location:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/doc/swd_svt_uvm_user_guide.pdf
```

2 Integrating the VIP into a User Testbench

The VC VIP for SWD 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 SWD Class Reference* (only in HTML format) at the following location:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/doc/
swd_svt_uvm_class_reference/html/index.html
```

2.1 VIP Testbench Integration Flow

The SWD agent (`svt_swd_master_agent` and `svt_swd_slave_agent`) is the top-level component provided by the VIP for modeling Transaction, Link and Physical layer. This generic agent encapsulates the following components:

- Sequencer
- Monitor
- Driver

Agents are instantiated and created in top-level environment and can be configured as Master or Slave by setting the `device_type` to `SWD_MASTER` or `SWD_SLAVE` in the SWD configuration class (`svt_swd_configuration`).

```
device_type = svt_swd_types::SWD_MASTER;
device_type = svt_swd_types::SWD_SLAVE;
```

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

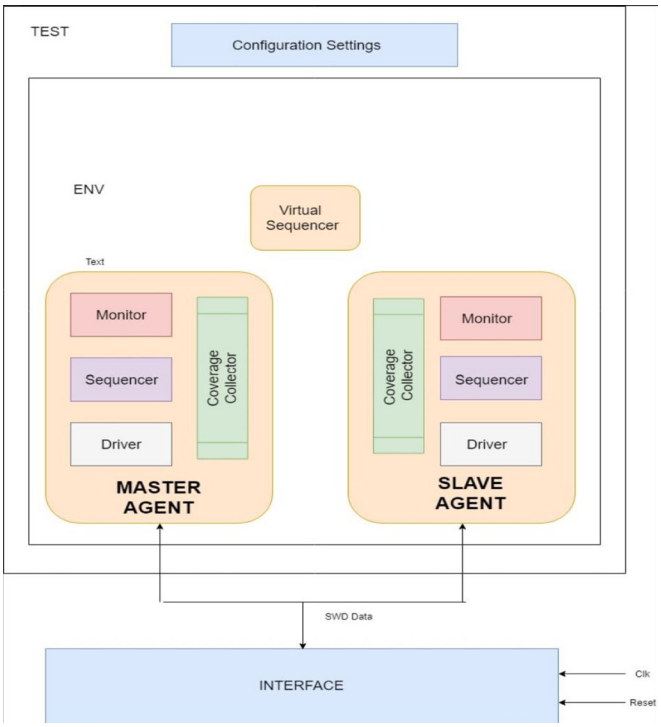


Figure 2-1 is a top-level architecture of a simple VC VIP for SWD testbench. The steps for integrating the VIP into a UVM testbench are described in the following sections:

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

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/swd_svt/latest/examples/sverilog/
tb_swd_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 "svt_swd.uvm.pkg" //Top-level VIP package
import uvm_pkg::*;
`include "uvm_macro.svh"
import svt_uvm_pkg::*;
import svt_swd_uvm_pkg::*; //UVM SWD package
```

- Instantiate the top-level SWD interface.

```
SWD Interface (Master)
svt_swd_if if_port ();
.clk_m
.resetn_m
.swd_data_m
(port_if.swd_clk ) ,
(port_if.reset) ,
(port_if.swd_data ),
Connect DUT's pins with SWD
SWD Interface(Slave)
svt_swd_if if_port1 ();
.clk_s
.resetn_s
.swd_data_s
(port_if1.swd_clk ) ,
(port_if1.reset) ,
(port_if1.swd_data ),
Connect DUT's pins with SWD
```

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

```
uvm_config_db#(virtual
svt_swd_if)::set(uvm_root::get(),"uvm_test_top.env.agent_bf m0*", "vif", if_port);
uvm_config_db#(virtual
svt_swd_if)::set(uvm_root::get(),"uvm_test_top.env.agent_bf m1*", "vif",
if_port1);
```

The `uvm_config_db` command connects the top-level SWD interface to the virtual interface of the SWD agent. The "`uvm_test_top`" represents the top-level module in the UVM environment. The `env` is an instance of SWD environment that encapsulates the SWD agent (`svt_swd_master_agent` and `svt_swd_slave_agent`).

2.1.2 Instantiating and Configuring the VIP

The following are steps to instantiate and configure the SWD agent (`svt_swd_master_agent` and `svt_swd_slave_agent`) in your testbench environment.

- Instantiate the SWD agent (`svt_swd_master_agent` and `svt_swd_slave_agent`) in the build phase of your testbench environment.

```
svt_swd_master_agent agent_bfm0;
svt_swd_slave_agent agent_bfm1 ;
agent_bfm0 = svt_swd_master_agent::type_id::create("agent_bfm0",this);
agent_bfm1 = svt_swd_slave_agent::type_id::create("agent_bfm1",this);
```

- Create a test configuration class by extending the SWD agent configuration class (`svt_swd_agent_configuration`). This configuration class has been provided for users to customize the configuration setting as per their requirements.

For example:

```
class cust_svt_swd_agent_configuration extends svt_swd_agent_configuration;

/** UVM object utility macro */
`uvm_object_utils(cust_svt_swd_agent_configuration)
/** Class constructor */
function new(string name
="cust_svt_swd_agent_configuration" );
super.new(name);
endfunction
endclass
```

For more information on the configuration class, refer to the *svt_swd_agent_configuration Class References* at the following location:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/doc/
swd_svt_uvm_class_reference/html/
class_svt_swd_agent_configuration.html
```

- Configure the VIP in the build phase of your testbench environment.

```
bfm_cfg = cust_svt_swd_agent_configuration::type_id::create("bfm_cfg");
bfm_cfg1 = cust_svt_swd_agent_configuration::type_id::create("bfm_cfg1");
```

- The `cust_svt_swd_agent_configuration` is the test configuration as defined in the previous step. The `bfm_cfg` and `bfm_cfg1` is an instance of this configuration.

2.1.3 Creating a Test

- You can create a base test class (`swd_base_test`) to specify the default test behaviors and to serve as the base class for other SWD tests. The following code snippets can be used in the base test class file (*swd_base_test.sv*).
- Instantiate the testbench environment and the test configuration, and construct these components in the build phase.

```
`include "swd_basic_env.sv" //Suggested UVM environment file
`include "cust_svt_swd_agent_configuration.sv"//configuration file
//Build phase
bfm_cfg = cust_svt_swd_agent_configuration::type_id::create("bfm_cfg");
env = swd_basic_env::type_id::create("env", this);
```

Examples from the VIP installation include a set of SWD tests. These tests are extended from the base test (`swd_base_test`) to create different test scenarios. For more information on the VIP tests, refer to the test files in the following directories:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/examples/sverilog/tb_swd_svt_uvm_basic_sys
```

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 SWD:

```
+define+SVT_UVM_TECHNOLOGY
+define+UVM_PACKER_MAX_BYTES=16384
+define+SVT_SWD_MAX_DATA_WIDTH=16 //User Specific Value to be used
+define+SVT_SWD_MAX_INSTRUCTION_WIDTH=8//User Specific Value to be used
```

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.

For VCS simulations, the following timescale specification is required.

```
-timescale=1ps/1fs
```

Macro	Description
SVT_UVM_TECHNOLOGY	Specifies SystemVerilog based VIPs that are compliant with UVM
UVM_PACKER_MAX_BYTES	Sets to 16384 or greater
SYNOPSYS_SV	Specifies SystemVerilog based VIPs that are compliant with UVM.

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=directed_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 SWD

VC Verification IP UVM Installation and Setup Guide:

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

VC VIP SWD UVM User Guide:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/doc/swd_svt_uvm_user_guide.pdf
```

VC VIP SWD Getting Started Guide:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/doc/
swd_svt_uvm_getting_started.pdf
```

VC VIP SWD Class Reference:

```
$DESIGNWARE_HOME/vip/svt/swd_svt/latest/doc/
swd_svt_uvm_class_reference/html/index.html
```

A.3 Example Home Directory

Directory that contains a list of VIP example directories:

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

View simulation options for each example:

```
gmake help
```