Verification Continuum™

VC Verification IP JTAG UVM Getting Started Guide

Version Q-2020.06, June 2020

SYNOPSYS®

Contents

Preface

About This Document

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

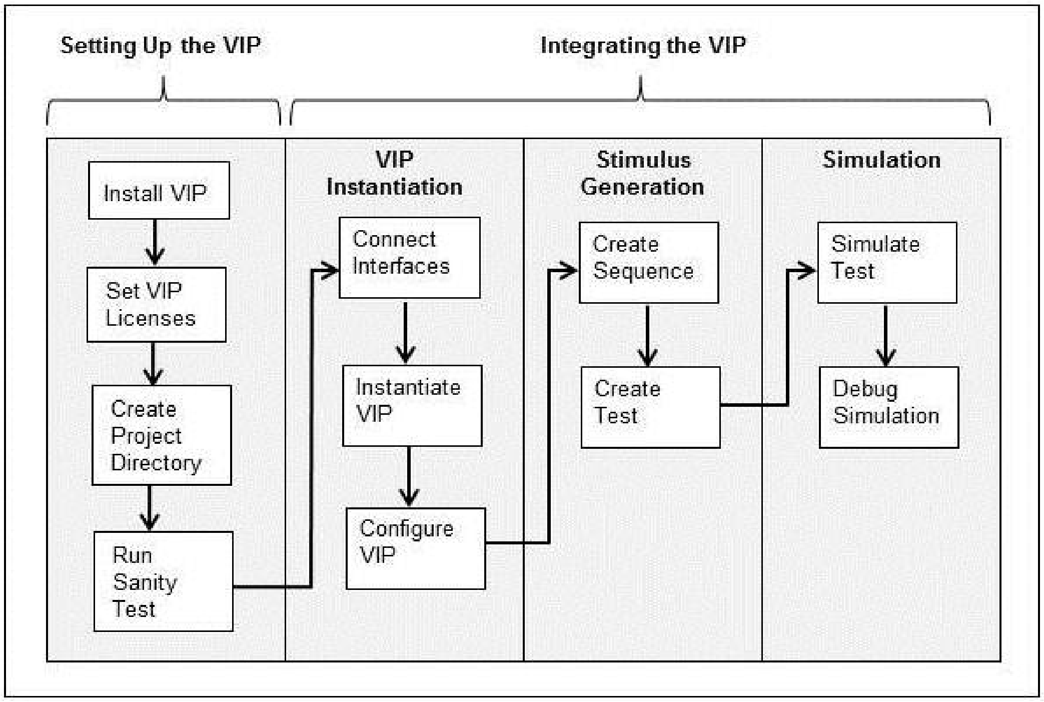
1 Overview of the Getting Started Guide

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

$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_uvm_user_guide.pdf

2 Integrating the VIP into a User Testbench

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

$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_uvm_class_reference/html/index.html

2.1 VIP Testbench Integration Flow

The JTAG agent (`svt_jtag_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

The JTAG agent can be configured as Driver or Controller by setting the `jtag_device_type` to `JTAG_DRIVER` or `JTAG_CONTROLLER` in the JTAG configuration class (`svt_jtag_configuration`).

```
        jtag_device_type = svt_jtag_types::JTAG_CONTROLLER;
        jtag_device_type = svt_jtag_types::JTAG_DRIVER;
```

You can instantiate and construct the JTAG agent in the top-level environment of your UVM testbench.

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

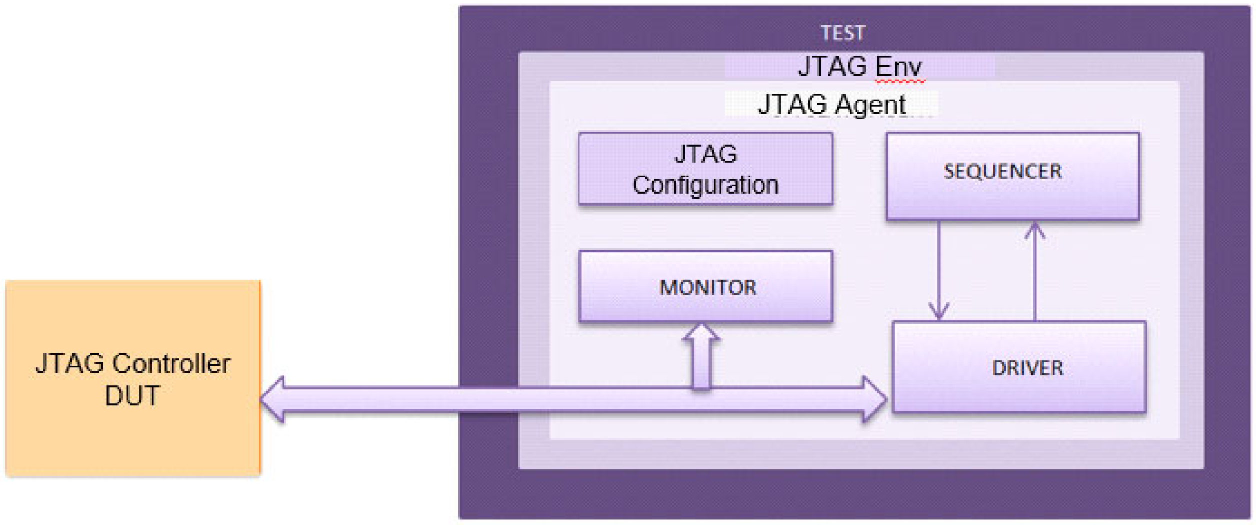


Figure 2-1 is a top-level architecture of a simple VC VIP for JTAG testbench. The steps for integrating the VIP into a UVM testbench are described in the following sections:

- "Connecting the VIP to the DUT"
- "Instantiating and Configuring the VIP"
- "Creating a Test"
- "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/jtag_svt/latest/examples/sverilog/tb_jtag_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_jtag.uvm.pkg" //Top-level VIP package

import uvm_pkg::*;
`include "uvm_macro.svh"
import svt_uvm_pkg::*;
import svt_jtag_uvm_pkg::*; //UVM JTAG package
```

- Instantiate the top-level JTAG interface.

```
svt_jtag_if if_port ();

  .tx_clk            (port_if.tck ),
  .tx_resetn         (port_if.trst),
  .tx_tdi            (port_if.tdi ),
  .tx_tms            (port_if.tms ),
  .tx_tdo            (port_if.tdo ),

Connect DUT's pins with JTAG
```

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

```
uvm_config_db#(virtual
svt_jtag_if)::set(uvm_root::get(),"uvm_test_top.env.agent_bfm0*",
"vif", if_port);
```

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

2.1.2 Instantiating and Configuring the VIP

The following are steps to instantiate and configure the JTAG agent (`svt_jtag_agent`) in your testbench environment.

- Instantiate the JTAG agent (`svt_jtag_agent`) in the build phase of your testbench environment.

```
svt_jtag_agent agent_bfm0;

agent_bfm0 = svt_jtag_agent::type_id::create("agent_bfm0",this);
```

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

  For example,

```
class cust_svt_jtag_agent_configuration extends
svt_jtag_agent_configuration;

 /** UVM object utility macro */
  `uvm_object_utils(cust_svt_jtag_agent_configuration)

 /** Class constructor */
  function new(string name ="cust_svt_jtag_agent_configuration"
);
    super.new(name);
  endfunction

endclass
```

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

$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_uvm_class_reference/html/class_svt_jtag_agent_configuration.html

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

```
bfm_cfg = cust_svt_jtag_agent_configuration::type_id::create("bfm_cfg");
```

The `cust_svt_jtag_agent_configuration` is the test configuration as defined in the previous step. The `bfm_cfg` is an instance of this configuration.

2.1.3 Creating a Test

You can create a base test class (`jtag_base_test`) to specify the default test behaviors and to serve as the base class for other JTAG tests. The following code snippets can be used in the base test class file (*jtag_base_test.sv*).

- Instantiate the testbench environment and the test configuration, and construct these components in the build phase

```
`include "jtag_basic_env.sv" //Suggested UVM environment file
`include "cust_svt_jtag_agent_configuration.sv" //Suggested test
//configuration file
//Build phase
bfm_cfg = cust_svt_jtag_agent_configuration::type_id::create("bfm_cfg");

env = jtag_basic_env::type_id::create("env", this);
```

Examples from the VIP installation include a set of JTAG tests. These tests are extended from the base test (`jtag_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/jtag_svt/latest/examples/sverilog/tb_jtag_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 JTAG:

```
+define+SVT_UVM_TECHNOLOGY
+define+UVM_PACKER_MAX_BYTES=16384
+define+SYNOPSYS_SV
+define+SVT_JTAG_MAX_DATA_WIDTH=16 //User Specific Value to be used
+define+SVT_JTAG_MAX_INSTRUCTION_WIDTH=8//User Specific Value to be used
+incdir+project_directory_path/vip/svt/common/latest/C/lib/platform/
libmemserver.so
```

Where, `project_directory_path` is your project directory and `platform` is linux, amd64 or suse64.

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 JTAG

VC Verification IP UVM Installation and Setup Guide:

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

VC VIP JTAG Release Notes:

$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_release_notes.pdf

This document includes the enhancements and list of STARs fixed in the current release and also includes the history of previous releases.

VC VIP JTAG UVM User Guide:

$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_uvm_user_guide.pdf

This document covers downloading and installing VIP, integration steps, list of VIP features, etc.

VC VIP JTAG FAQ:
$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_faq.pdf
This document covers the frequently asked questions.

VC VIP JTAG Getting Started Guide:
$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_uvm_getting_started.pdf

VC VIP JTAG Class Reference:
$DESIGNWARE_HOME/vip/svt/jtag_svt/latest/doc/jtag_svt_uvm_class_reference/html/index.html
This document includes the VIP's attributes, subroutines, coverbins, protocol checks, etc.

A.3 Example Home Directory

Directory that contains a list of VIP example directories:

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

View simulation options for each example:

```
gmake help
```