Clock Verification IP v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG291 (v1.0) October 30, 2019

XILINX®

Table of Contents

IP Facts

The Xilinx® Clock Verification IP (VIP) core has been developed to support the simulation of customer designed test bench or design which requires a clock signal.

The Clock VIP is unencrypted SystemVerilog source that is comprised of a SystemVerilog interface and synthesizable RTL. You can use APIs from the embedded clock RTL interface to set up the clock period.

Features

- Sets interface into master/pass-through mode
- Starts/stops clock
- Sets/gets clock initial value
- Sets/gets clock period
- Sets/gets clock frequency
- Power ON/OFF clock jitter

IP Facts

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family[(1)]	UltraScale+, UltraScale, Zynq®-7000 SoC, 7 series
Supported User Interfaces	Clock
Resources	N/A
Provided with Core	
Design Files	N/A
Example Design	SystemVerilog
Test Bench	N/A
Constraints File	N/A
Simulation Model	Unencrypted SystemVerilog

LogiCORE™ IP Facts Table	
Supported S/W Driver	N/A
Tested Design Flows[2]	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	N/A
Support	
Release Notes and Known Issues	Master Answer Record: 69565
All Vivado IP Change Logs	Master Vivado IP Change Logs: 72775
Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of third-party tools, see the Xilinx Design Tools: Release Notes Guide.

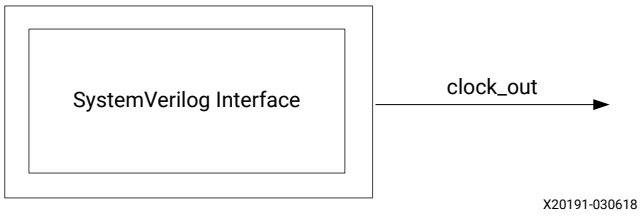
Overview

The Clock VIP core generates different kinds of clock signals during simulation. The Clock VIP can be configured in two different modes:

- Clock master VIP
- Clock pass-through VIP

The following figure shows the Clock master VIP which generates a clock signal and sends it to the clock system.

Figure 1: **Clock Master VIP**



The following figure shows the Clock pass-through VIP passing the clock signal which it receives. It can be configured in simulation to be pass-through or master.

Figure 2: **Clock Pass-Through VIP**

Feature Summary

The Clock VIP core can be configured in master or in pass-through mode.

Applications

The Clock VIP core is for verification and system engineers who want to generate clock signals.

Licensing and Ordering

This Xilinx® LogiCORE™ IP module is provided at no additional cost with the Xilinx Vivado® Design Suite under the terms of the Xilinx End User License.

Information about other Xilinx® LogiCORE™ IP modules is available at the Xilinx Intellectual Property page. For information about pricing and availability of other Xilinx® LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

This chapter includes information on performance, parameters, and port descriptions.

Performance

The Clock VIP core synthesizes to wires and does not impact performance.

User Parameters

The following table shows the Clock VIP core user parameters.

Table 1: **Clock VIP User Parameters**

Parameter Name	Format/Range	Default Value	Description
INTERFACE_MODE	Type: string Value range: PASS_THROUGH, MASTER	PASS_THROUGH	Used to control the mode of protocol to be configured as master or pass-through.
FREQ_HZ	Type: float Value range: 1, 1,000,000,000	100000000	Used to control the default setting of clock frequency.

Port Descriptions

The table shows the Clock VIP independent port descriptions.

Table 2: **Clock VIP Independent Port Descriptions**

Signal Name	I/O	Default	Width	Description	Enablement
clk_in	I		1	Clock input	In pass-through mode only
clk_out	O		1	Clock output	Always ON

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

General Design Guidelines

The Clock VIP core should be inserted into a system as shown in the following figures for Clock master VIP and Clock pass-through VIP.

Figure 3: **Clock Master VIP Example Topology**

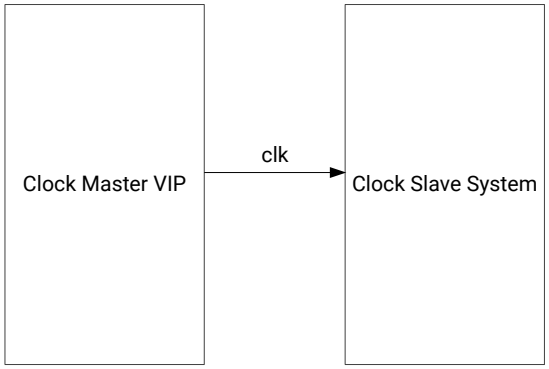
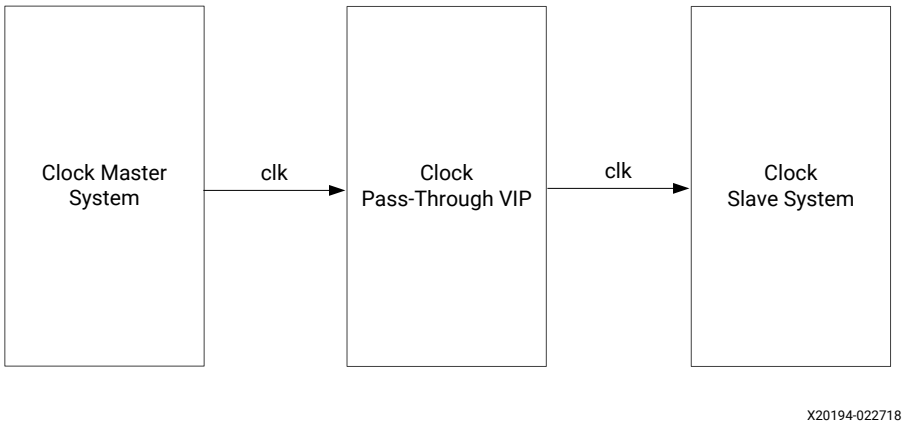


Figure 4: **Clock Pass-Through VIP Example Topology**



***Note*:** When the Clock VIP is configured with Asynchronous mode set to NO, `sync_clk` is added to the figures below.

Clocking

This section is not applicable for this IP core.

Resets

This section is not applicable for this IP core.

Design Flow Steps

This section describes customizing and generating the core, constraining the core, and the simulation, synthesis, and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
- *Vivado Design Suite User Guide: Designing with IP* (UG896)
- *Vivado Design Suite User Guide: Getting Started* (UG910)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado® Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

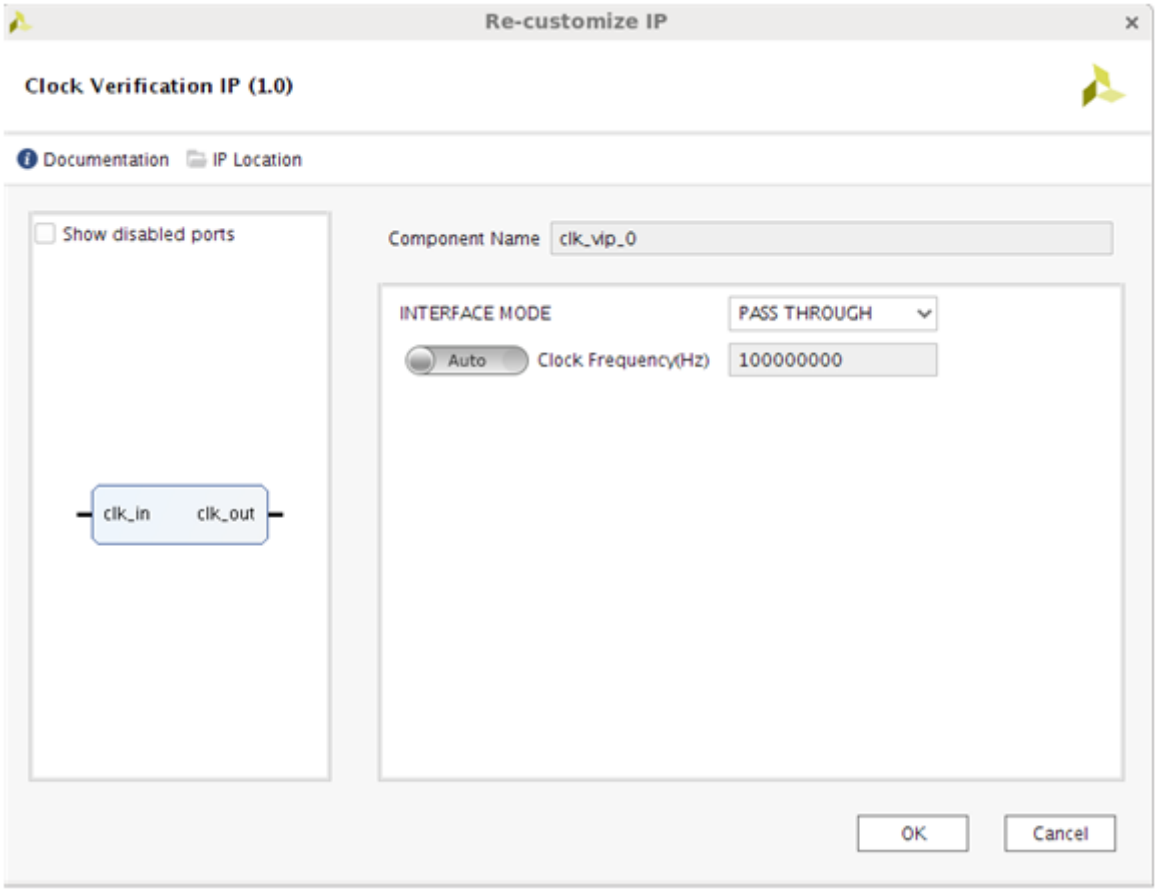
For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) and the *Vivado Design Suite User Guide: Getting Started* (UG910).

Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

Customize IP Window

The figure shows the Clock VIP Vivado IDE Component Name screen.

Figure 5: **Customize IP Window**



Note: For the runtime parameter descriptions, see the User Parameters table in the Product Specification.

- **Component Name:** The component name is used as the base name of output files generated for the module. Names must begin with a letter and must be composed from characters: a to z, 0 to 9 and "_".
- **Interface Mode:** Controls the mode of protocol to be configured as master or pass-through.
- **Clock Frequency:** Selects the specific clock frequency specification.

User Parameters

For the relationship between the fields in the Vivado IDE and the User Parameters (which can be viewed in the Tcl Console), see the User Parameters table in the Product Specification chapter.

Related Information
User Parameters

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896).

The Clock VIP core deliverables are organized in the directory `<project_name>/<project_name>.srcs/sources_1/ip/<component_name>` and are designated as the `<ip_source_dir>`. The relevant contents or directories are described in the following sections.

Vivado Design Tools Project Files

The Vivado design tools project files are located in the root of the `<ip_source_dir>`.

Table 3: **Vivado Design Tools Project Files**

Name	Description
<component_name>.xci	Vivado tools IP configuration options file. This file can be imported into any Vivado tools design and be used to generate all other IP source files.
<component_name>.{veo|vho}	Clock VIP instantiation template.

IP Sources

The IP sources are held in the subdirectories of the `<ip_source_dir>`.

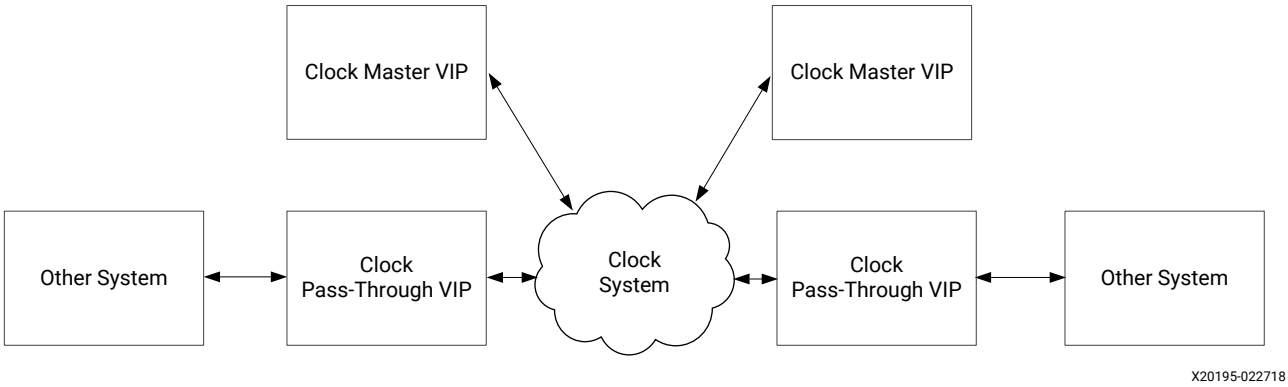
Table 4: **IP Sources**

Name	Description
hdl/*.sv	Clock VIP source files.
synth/<component_name>.sv	Clock VIP generated top-level file for synthesis. Optional, generated if synthesis target selected.
sim/<component_name>.sv	Clock VIP generated top-level file for simulation. Optional, generated if simulation target selected.

Clock VIP in Vivado IP Integrator

This section contains information about how to use the Clock VIP in a design and test bench environment. The following figure shows a possible design with the Clock VIPs.

Figure 6: **Clock VIP Design**



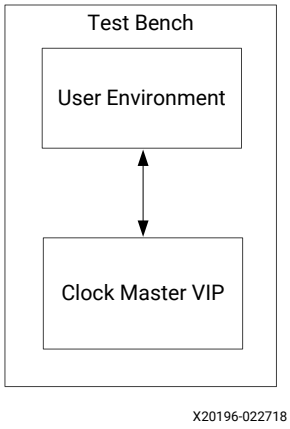
The Clock VIP consists of an interface which is used to generate clock signal which is needed in any design.

Clock Master VIP

The figure shows the Clock master VIP with its test bench. The test bench has two parts:

- User environment
- Clock master VIP

Figure 7: **Clock Master VIP Test Bench**



Finding the Clock VIP Hierarchy Path in IP Integrator

As mentioned earlier, the Clock VIP interface has to be passed to the user environment for use. The following guidelines describe how to find the hierarchy path of the Clock VIP in the IP integrator.

The best method to identify the VIP instance in the hierarchy is after the connection of all the IPs and the validation check. Click the **Simulation Settings**, set up the tool, and then click **Run Simulation**. The figure shows the Mentor Graphics Questa Advanced Simulator results. After the hierarchy is identified, it is used in the SystemVerilog test bench to drive the Clock VIP APIs.

Figure 8: **Clock VIP Instance in IP Integrator Design Hierarchy**

After the Clock VIP is instantiated in the IP integrator design and its hierarchy path found, the next step is using the Clock VIP to generate clock in master mode or produce pass-through VIP in runtime master mode.

Related Information

Example Design

Constraining the Core

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

For comprehensive information about Vivado® simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900).

IMPORTANT! *For cores targeting 7 series or Zynq®-7000 devices, UNIFAST libraries are not supported. Xilinx IP is tested and qualified with UNISIM libraries only.*

Synthesis and Implementation

The Clock VIP core is a verification IP set to synthesize as wires. There is no implementation for the Clock VIP.

Chapter 6

Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite.

IMPORTANT! *The example design of this IP is customized to the IP configuration. The intent of this example design is to demonstrate how to use the Clock VIP core.*

Overview

The following figure shows the Clock VIP core example design.

Figure 9: **Clock VIP Example Design**

This section describes the example tests used to demonstrate the abilities of the Clock VIP core. Example tests are delivered in SystemVerilog. When the core example design is open, the example files are delivered in a standard path test bench and bd design are under directory imports. The packages are under the directory `example.srcs/sources_1/bd/ex_sim/ipshared`.

The example design consists of two components:

- Clock VIP in master mode
- Clock VIP in pass-through mode

In the Clock master VIP, it creates a clock signal and sends it to the Clock pass-through VIP. In the Clock pass-through VIP, it receives a clock signal from the Clock master VIP and sends it out.

The Clock VIP core is not fully autonomous. If the tests are written using the APIs, there are different methods from the user environment to set up the clock signal such as clock initial value, clock period, clock jitter range, etc. Xilinx recommends obtaining all of the members through the APIs instead of accessing them directly.

When the Clock VIP is configured in pass-through mode, it can be changed to master mode in the runtime and then be changed back to pass-through mode based on your requirements. When it is switched to runtime master mode, it behaves exactly as a Clock master VIP.

Test Bench

This chapter contains information about the test bench for the example design provided in the Vivado® Design Suite.

To open the example design either from the Vivado IP catalog or Vivado IP integrator design, follow these steps:

1. Open a new project and click **IP Catalog**.
2. Search for **Clock Verification IP**. Double-click the IP, configure, and generate the IP.
3. Right-click the IP and choose **Open IP Example Design...**.

   ***Note*:** If you have the Clock VIP as one component in the IP integrator design, right-click Clock VIP and click **Open IP Example Design...**.

In both scenarios, a new project with the example design is created. The example design has the master and pass-through VIP connected directly to each other as shown in the figure in the Example Design chapter. The configuration of the example design matches the original VIP configuration.

Related Information

Example Design

Clock VIP Example Test Bench and Test

The following scenarios are covered in the example design:

- Clock pass-through VIP in pass-through mode. The Clock master VIP generates a simple clock signal and passes it to the pass-through VIP.
- Switches Clock pass-through VIP into the runtime master mode and generates a simple clock signal.

Useful Coding Guidelines and Examples

While coding test bench for the Clock VIP, the following requirements must be met. Otherwise, the Clock VIP does not function.

1. Create module test bench as all other standard SystemVerilog test benches.

```
module testbench();
      …
      endmodule
```

2. To start the clock generation, use `<hierarchy_path>.IF.start_clock`.
3. To stop the clock generation, use `<hierarchy_path>.IF.stop_clock`.
4. To change clock period from default to new, use the following code block:

```
<hierarchy_path>.IF.set_clk_prd or set_clk_frq. set_clk_prd sets the
clock period, duty_cycle, jitter on/off, minimum jitter, and
maximum jitter. Function: set_clk_frq sets clock frequency, duty_cycle,
jitter on/off, minimum jitter, and
maximum jitter.
```

5. You can use `set_initial_value()` to set the clock initial value in the Clock VIP.
6. APIs used to switch pass-through VIP into runtime master and runtime pass-through modes are `set_master_mode` and `set_passthrough_mode`.
7. Use the following code to switch the Clock pass-through VIP into the runtime master mode. The `<hierarchy_path>` can be found in Finding the Clock VIP Hierarchy Path in IP Integrator section:

```
< hierarchy_path>.set_master_mode();
    < hierarchy_path>.IF.start_clock();
```

8. Use the following code to switch the Clock pass-through VIP into the runtime pass-through mode. The `<hierarchy_path>` can be found in Finding the Clock VIP Hierarchy Path in IP Integrator section:

```
< hierarchy_path>.set_passthrough_mode();
```

9. The following are example codes for the Clock VIP:

```
clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_mst.inst.IF.start_clock();

    #(10*10*1ns);
    /
**************************************************************************
******************
    *    Update clock by calling set_clk_prd with clock period, duty
cycle,jitter on/off, if jitter is on,
    *    its value will randomly pick from minimum jitter range and
maximum jitter range.
    *    Turn on the monitor and check expected clock period, duty
cycle, and jitter.
```

```
*****************************************************************************
******************/
    clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_mst.inst.IF.set_clk_prd(
    .user_period(1000),
    .user_duty_cycle(0.4),
    .user_jitter_on(1),
    .user_jitter_min_range(0.0),
    .user_jitter_max_range(0.01)
    );

    #4000ns;
    /
*****************************************************************************
******************
    *     Update clock by calling set_clk_frq with clock frequency, duty
cycle,jitter on/off,
    *     if jitter is on, its value will randomly pick from minimum
jitter range and maximum jitter range.
    *     Turn on the monitor and check expected clock period, duty
cycle and jitter.

*****************************************************************************
******************/
    clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_mst.inst.IF.set_clk_frq(
    .user_frequency(10000000),
    .user_duty_cycle(0.2),
    .user_jitter_on(0),
    .user_jitter_min_range(0.0),
    .user_jitter_max_range(0.00)
    );
    #400ns;
    /
*****************************************************************************
******************
     update clock again


*****************************************************************************
******************/
    clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_mst.inst.IF.set_clk_prd(
    .user_period(200),
    .user_duty_cycle(0.3),
    .user_jitter_on(1),
    .user_jitter_min_range(0.0),
    .user_jitter_max_range(0.01)
    );

    /
*****************************************************************************
******************
     stop clock and disable master monitor

*****************************************************************************
******************/
    clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_mst.inst.IF.stop_clock();

    #(1*1ns);
    /
*****************************************************************************
******************
    switch passthrough VIP into runtime master mode
      *      start clock-- this will generate the default clk with
```

```
period(default passthrough clk
*          vip period) with no jitter,  duty cycle 50%
*       turn on mst_monitor clk check with expected clk settings


*********************************************************************
******************/

clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_passthrough.inst.set_master_mod
e();
    #1ns;

clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_passthrough.inst.IF.set_initial_
value(0);

clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_passthrough.inst.IF.start_clock(
);
    #(10*10*1ns);
    /
*********************************************************************
******************
    *     update clock by calling set_clk_prd with clock period, duty
cycle,jitter on/off,
    *     if jitter is on, its value will randomly pick from minimum
jitter range and maximum jitter range.
    *     Turn on the monitor and check expected clock period, duty
cycle and jitter.


*********************************************************************
******************/

clk_vip_0_exdes_tb.DUT.ex_design.clk_vip_passthrough.inst.IF.set_clk_prd(
    .user_period(1000),
    .user_duty_cycle(0.3),
    .user_jitter_on(1),
    .user_jitter_min_range(0.0),
    .user_jitter_max_range(0.01)
    );
```

Related Information

Finding the Clock VIP Hierarchy Path in IP Integrator

Example Design

Appendix A

Upgrading

This appendix is not applicable for the first release of the core.

Clock VIP APIs

This appendix contains information about the `clk_vip_v1_0_top` APIs. These APIs can be called through the following code. The `set_passthrough_mode` and `set_master_mode` are used to switch the pass-though VIP into different runtime modes. These APIs can be called through the test bench hierarchy pointing to the top. An example would be `set_passthrough_mode()`.

```
<hierarchy_path>.set_passthrough_mode()

set_passthrough_mode
function void set_passthrough_mode()
//Sets CLK VIP passthrough into run time passthrough mode


set_master_mode
function void set_master_mode()
//Sets CLK VIP passthrough into run time master mode
```

Related Information

Finding the Clock VIP Hierarchy Path in IP Integrator

Appendix C

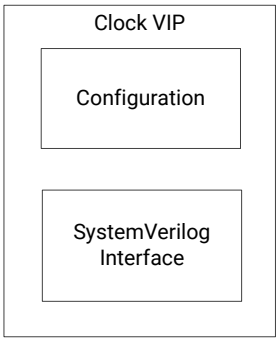
Clock VIP Generation and Flow Methodology

This appendix contains information about the Clock VIP agents and flow methodologies.

Core Architecture

Before talking about how to use Clock VIP core, the VIP architecture is described here. Different from other standard Xilinx IP, the Clock VIP core is based on the SystemVerilog interface. The Clock VIP core architecture is shown here.

Figure 10: **Clock VIP Core Architecture**

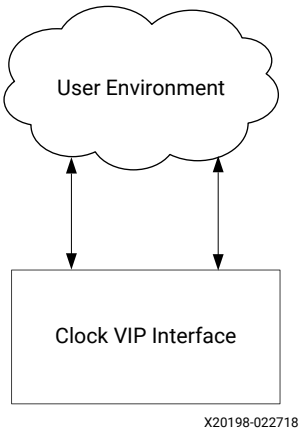


The Clock VIP core consist of two main layers:

- SystemVerilog signal interface
- Configuration

The SystemVerilog signal interface includes the typical Verilog input/output ports which are `clk_in` and `clk_out`. You can access the available APIs to set the clock initial value, clock period, and clock jitter. For more information about usage and list of APIs in the Clock VIP, see the API documentation.

Figure 11: **Clock VIP Interface**



Clock VIP Generation Flow

The following steps outline how to generate a clock using the Clock VIP core.

1. To generate the clock, call `<hierarchy_path>.IF.start_clock`.
2. To stop clock generation, call `<hierarchy_path>.IF.stop_clock`.
3. To generate a different clock, you can call `<hierarchy_path>.IF.set_initial_value` to set up the clock initial value.
4. There are two APIs that set up the clock period, duty cycle, jitter ON or OFF, jitter minimum range, jitter maximum, and jitter range.
   - The first is called `set_clk_prd(user_period, user_duty_cycle, user_jitter_on, user_jitter_min_range, user_jitter_max_range)`.
   - The second is called `set_clk_frq(user_frequency, user_duty_cycle, user_jitter_on, user_jitter_min_range, user_jitter_max_range)`.
5. If the Clock VIP is in pass-through mode, switch its mode to be in runtime master by calling `<hierarchy_path>.set_master_mode`.

Appendix D

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools.

Finding Help on Xilinx.com

To help in the design and debug process when using the core, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support. The Xilinx Community Forums are also available where members can learn, participate, share, and ask questions about Xilinx solutions.

Documentation

This product guide is the main document associated with the core. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx® Documentation Navigator. Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use keywords such as:

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master AR for Core

AR 69565

Technical Support

Xilinx provides technical support on the Xilinx Community Forums for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To ask questions, navigate to the Xilinx Community Forums.

XILINX

Appendix E

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Getting Started* (UG910)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
5. Clock VIP API Documentation

   Note: VIP API documentation source codes are different from the Install area implementation codes, refer to the Install area for the source codes.

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
10/30/2019 Version 1.0	
References	Updated VIP API documentation link.
05/22/2019 Version 1.0	
References	Added VIP API documentation link.
11/14/2018 Version 1.0	
Document updates.	Updated to latest Vivado Design Suite release.
04/04/2018 Version 1.0	
Initial release.	N/A

Please Read: Important Legal Notices

specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright