Verification Continuum™

VC VIP USB
Performance Metrics
Supported Through Verdi

Version V-2023.09, September 2023

SYNOPSYS®

Preface

About This Manual

This manual contains installation, setup, and usage material for SystemVerilog UVM users of the Discovery USB VIP, and is for design or verification engineers who want to verify USB operation using a UVM testbench written in SystemVerilog. Readers are assumed to be familiar with USB, Object Oriented Programming (OOP), SystemVerilog, and Universal Verification Methodology (UVM) techniques.

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:

1. 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.

2. 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.

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.

Performance Metrics

The following is the list of USB Performance Metrics and its description:

Table 3-1   USB Metrics Description

Metrics	Description
usb_avg_trans_bus_transfer_rate	This metric computes the average transfer rate for completed transaction.
usb_lup_ldn_time	This metric computes the begin time of every LUP/LDN.
usb_packet_ss_tp_distribution	This metric computes the Transaction Packets.
usb_transaction_ss_distribution	This metric computes all the SS/SSP Transactions.
usb_avg_trans_link_response_latency	This metric computes the average of time between the SS Transaction neing sent and correcponding Link Command response received.
usb_max_trans_bus_transfer_rate	This metric computes the maximum transfer rate per completed transaction.
usb_previous_pkts_lcmds_first_lup_ldn	This metric computes the time taken to send LUP/LDN after the last Packet/LinkCommand.
usb_trans_bus_transfer_rate	This metric computes the rate at which each transfer of a transaction happens.
usb_cinst_total_transfered_bytes	This metric computes the total bytes trasfered.
usb_max_trans_link_response_latency	This metric computes the maximum of all the time between the each SS Transaction neing sent and correcponding Link Command response received.
usb_rx_detect_active_to_polling_lfps_latency	This metric computes the time taken for the SS LTSSM to move from Rx.Detect.Active to Polling.
usb_transfer_20_distribution	This metric computes all the 2.0 Transfers.
usb_lfps_first_tx_last_rx_latency	This metrci computes the time between the first LFPS sent and the first LFPS received after one being sent.
usb_min_trans_bus_transfer_rate	This metric computes the minimum of all the transfers in a transaction.
usb_symbol_set_rx_tx_count	This metric computes the number of Symbols Sets transmitted and received.

Table 3-1   USB Metrics Description

Metrics	Description
`usb_transfer_bulk_in_latency`	This metric computes the time taken to complete a Bulk in Transfer.
`usb_lfps_recieve_transmit_count`	This metric computes the number of LFPSs sent and received.
`usb_min_trans_link_response_latency`	This metric computes the minimum of all time taken between a Transfer and the Link Command received.
`usb_transaction_20_distribution`	This metric computes the number of all the transaction of USB 2.0.
`usb_transfer_isoc_in_latency`	This metric computes the time of each ISOC transfer of USB SS.
`usb_lfps_rx_tx_latency`	This metric computes the time between the first LFPS being sent and the first LFPS being received.
`usb_packet_20_distribution`	This metrci computes the number of Packets for USB 2.0.
`usb_transaction_in_latency`	This metric computes the time taken for each IN Transaction for USB SS.
`usb_transfer_ss_distribution`	This metric computes all the USB SS transfers.
`usb_lgo_to_lau_latency`	This metric computes the time taken between the LGO Link command sent and the time taken for LAU/LXU Link Command received.
`usb_packet_ss_distribution`	This metric computes the count of all differen types of USB SS Packets.
`usb_transaction_out_latency`	This metric computes the time taken for each OUT Trasactions of USB SS.
`usb_trans_link_response_latency`	This metric computes the time taken between the Transaction sent and Link Command response received in USB SS.