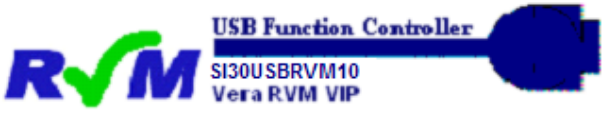
Factsheet SI30USBRVM10

Silicon Cores™

Core to the Intelligent System™

USB 2.0 Vera RVM VIP



Silicon Interfaces' **USB 2.0 Vera RVM VIP is fully documented, off the shelf component for the Verification of the USB 2.0 compliant Function Controller.**

USB 2.0 Vera RVM VIP can work in a standalone mode i.e. can be plugged with any Function Controller with standard pinouts without disturbing the structure.

This VIP is developed using the Synopsys' Vera Reuse Verification Methodology that is used in dynamic simulation of USB 2.0 based design.

The VIP provides a fast, accurate way to simplify and speed-up the device verification task. In a complex design process, verification may take up to 70% of the development time. USB 2.0 Vera RVM VIP speeds up the verification process providing a compelling cost and time to market.

Object Oriented Programming approach plays one of the key roles to achieve these goals. Writing a reusable code makes it easy for the Verification Engineer to apply the same tasks in various modules from project to project. The code becomes maintainable. Lots of efforts and time is saved. This ultimately yields an efficient and more productive solution with a shorter turn-around time.

Product Specifications

- The VIP can work with **8-bit or 16-bit** standard USB devices
- **Error Injection Mechanism**, which can be turned ON or OFF for a given simulation run. Incorporated around 45+ scenarios for different Error types
- Provides a choice for **RESETTING** to either **High-Speed** or **Full-Speed** mode upon startup
- **Supports RESET / SUSPEND / RESUME**. On-the-fly Reset switch over from High-Speed to Full-Speed or vice-versa supported

Product Highlights

- ☑ Synopsys RVM compliant
- ☑ Absence of inter-module dependencies makes it highly **reusable**
- ☑ Following **USB 2.0** Modes are being supported:
  - Full-Speed / High-Speed - 8-bit
  - Full-Speed / High-Speed - 16-bit
  - Low-Speed-only
  - Full-Speed-only
- ☑ Programmable and Randomized Transfer Types, namely, **INTERRUPT**, **ISOCHRONOUS**, **BULK** and **CONTROL** Transfer Types
- ☑ **Auto Programming** – cum selection of Clocks for different Modes – 60 MHz (Low-Speed-only Mode), 30 MHz (16-bit High-Speed / Full-Speed Mode), 48 MHz (Full-Speed-only Mode) or 60 MHz (8-bit High-Speed / Full-Speed Mode)
- ☑ **Hierarchical Seed Randomization**. Common Seed to all Randomized tasks.
- ☑ Fully **Constrained Driven Randomization** of Packets achieved by Randomization of various fields of the Packet
- ☑ Presence of a DUT **Health check-up mechanism** during initialization
- ☑ Provides a choice for inhibiting **SUSPEND** and / or **RESET** operation through use of simple macros
- ☑ Supports up to **16 Endpoints** as per the **USB 2.0** Standard. Endpoints can be either **IN**, **OUT** or **NOT_IMPLEMENTED** through user-defined header declarations, with Endpoint 0 as INOUT
- ☑ The product is bundled with a very fancy, intelligent and versatile **Scoreboard**
  - Provides a bird's eye view of the **USB** Transactions
  - Total number of Transactions
  - Statistics of Transfer types
  - Count of Token, Data and Handshake Packets
  - Statistics on **RESET** / **SUSPEND** / **RESUME** activities
  - Error Injection statistics
  - Overall count of good / bad Packets

606, Nirman Kendra, E. Moses Road, Mahalaxmi, Mumbai 400 011 India
www.siliconcores.com and www.siliconinterfaces.com

Layered Architecture:

In the RVM Test Environment, each layer provides a set of services to the upper layers, while abstracting it from the lower level details.

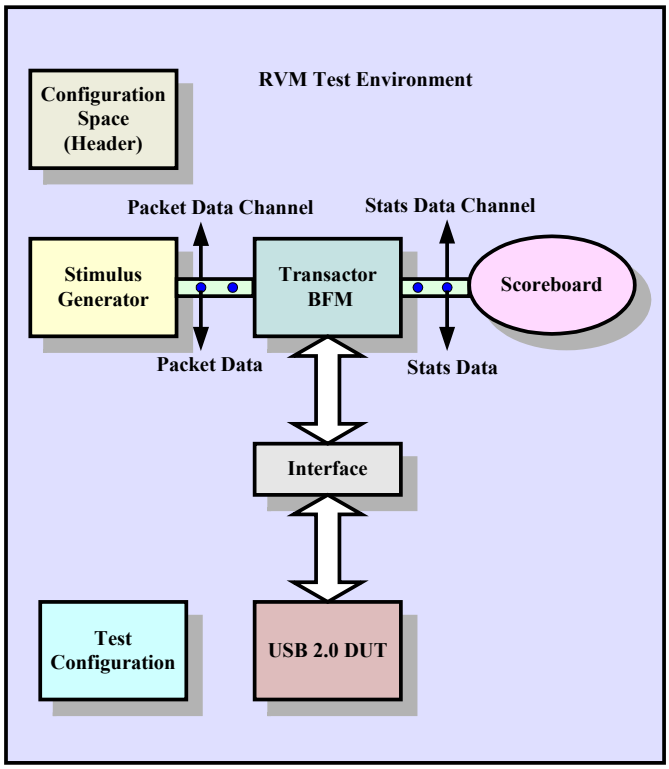
Signal Layer: This layer provides Signal-Level Connectivity in the physical representation of the DUT. This layer provides Signal Name Abstraction and connectivity to the event driven world of most simulation engines.

Command Layer: The Command Layer typically contains Bus-Functional Models, Physical-Level Drivers and Monitors associated with the various Interfaces and Physical-Level Protocols present in the DUT. It provides a consistent, Low-Level Transaction Interface to the DUT, regardless of how the DUT is modeled.

Functional Layer: The Functional Layer provides the necessary Abstraction Layers to process Application-Level Transactions and verify the correctness of the DUT.

Test Layer: Test Layer provides additional testcase-specific self-checking not provided by the Functional Layer at the Transaction Level. It can also perform checks where correctness depends on timing with respect to a particular Synchronization Event introduced by the testcase

USB 2.0 RVM Test Environment



Block Diagram Explanation

Stimulus Generator: This generates the Stimulus Packets. These Packets are constraint driven randomized raw data that is passed to the Transactor.

Transactor: It determines the sequence of the Packets and also the type of Packet to be sent to the DUT and receives response from the DUT. The Transactor also injects error in the Packets. It collects Transaction information and passes it to the Scoreboard.

Scoreboard: This receives information from the Transactor. Based on this information, it keeps a count of the types of Transaction and the various attributes of the Transactions. It also keeps a count of the number of IN Tokens, OUT Tokens, Data Packets etc

Interface Block: The Interfaces Block provides an Interface between the OpenVera testbench and the USB 2.0 DUT.

DUT: The DUT used here is the USB 2.0 Function Controller. The DUT supports both 8-bit and 16-bit modes in Low-Speed, Full-Speed as well as High-Speed

Contact:
Email: info@siliconinterfaces.com
Phone: (+91-22) 2491 3024; Fax: (+91-22) 2498 1379
In USA: (+1-408) 866 2458; Fax: (+1-408) 866 6586
In UK: (+44-20) 8543 4436; Fax: (+44-20) 8544 1311

***Silicon Interfaces*®**

a software and vlsi design center

***Silicon Interfaces* is the world's answer to integrated solutions for Design Services needs for SOC, ASIC & FPGA for Wireless, Cable & Optical Networking, Fiber Channel & Inter-connect, Data Communications and Storage. *Silicon Interfaces* is operating since 1990 primarily in USA, Europe and Pacific Rim.**