

# USBeVC

## Automated, Coverage-Driven Verification IP

### USBeVC

- ✓ USB Specification 1.1 and 2.0
- ✓ Supports USB On-the-Go (OTG)
- ✓ Functions as one or both:
  - USB Host
  - USB Device
- ✓ Runs at high, full or low speed
- ✓ Supports up to 127 devices
- ✓ Supports UTMI and UTMI+ interfaces
- ✓ Supports both hot configuration and backdoor configuration
- ✓ Supports all standard device requests and descriptors; includes infrastructure for class requests
- ✓ Total control over traffic generation using sequence interface including injection of errors
- ✓ Supports power signaling (single device configuration)
- ✓ Drives and responds to transactions in both random and directed fashion
- ✓ Built-in bus traffic monitors
- ✓ Support for coverage driven verification including built-in coverage analysis for packets and transactions
- ✓ Scoreboard checking of input/output
- ✓ HDL independent

eVerification Component Overview (eVC)  
 eVerification Components are reusable, configurable, pre-verified, plug-and-play verification environments. They offer the easiest to use, most complete module, chip and system level verification solution available. eVCs integrate automatic stimulus generation, assertion checking, and functional coverage analysis all within a single, extensible component. eVCs drastically reduce the time needed to compose a verification environment.

The philosophy underlying eVCs differs significantly from alternative products. Rather than use thousands of directed tests, the eVC employs automatic generation and a coverage driven methodology. Using automated scenario generation the eVC can typically achieve 90-95%+ coverage of the protocol. With the addition of a few tests the remaining corner cases are then exercised. This approach uncovers more bugs faster and frees engineering time to focus on testing the Device Under Test (DUT) proprietary functionality.

### Quality and Productivity Gains

With eVCs verification environments are created in days instead of weeks or months. You can begin running tests much earlier and achieve a much higher quality product.

Furthermore, eVerification Components can be reused without expending any extra effort. This enables you to retain your investment when moving from module to system level verification as well as when verifying derivative products.

### USBeVC Overview

The USBeVC verifies designs that include the USB 2.0 or 1.1 protocol. Verisity has worked carefully along with our numerous USB/C users to ensure that the USBeVC accurately verifies the USB protocol and thoroughly exercises the DUT. The eVC is mature and robust. It has verified over 80 designs for over 35 different design teams.

### Configurable Verification Environment

Verisity's USBeVC enables you to create many different configurations of your verification environment. Two examples are:

### Verifying a USB device

An eVC host initiates transactions both to the DUT device and to an eVC device. At all times, the eVC monitors the DUT USB interface signals and captures the bus traffic into USB abstract data items such as packets, transactions and transfers. Captured data is checked to verify adherence with the USB protocol and sampled by the eVC coverage engine to indicate progress in your verification task. You can also browse the data.

# USB eVC

## Verisity – Meet your SpeX

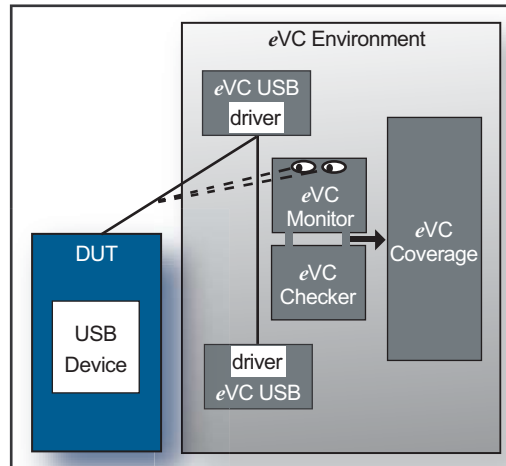
### Contact Information

Verisity Design, Inc.  
 331 East Evelyn Avenue  
 Mountain View, CA 94041  
 PH: (650) 934-6800  
 FX: (650) 934-6801  
 www.verisity.com

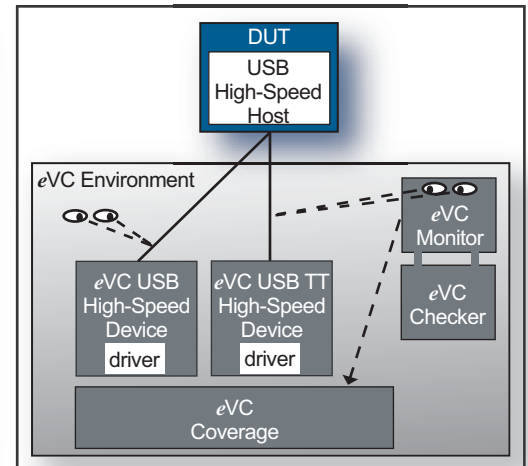


### For More Information

To find out more about the USB eVC contact your Verisity account manager or see us on the web at [www.verisity.com](http://www.verisity.com).



Verifying a USB Device



Verifying a USB Host Running at High Speed

### Verifying a USB host running at high speed

The eVC instantiates two high-speed devices including a normal device and a Transaction Translator (TT). The TT emulates a connection to a full- or low-speed device. The traffic monitored, covered and checked is that which goes into the DUT host.

### Deliverables

- Fully verified USB eVC code in e
- Documentation, including—user guide and release notes
- Standalone introductory demonstration
- Sample, extensible tests covering basic functionality

### USB eVC Functional Description

Host	Active emulation of USB host that generates and drives data onto the bus
Device	Active emulation of a USB device that appropriately responds to host transactions
Monitor	Passive component to monitor, check and collect functional coverage of bus traffic

*Protocol checking and coverage collecting require the monitor element. Otherwise, each of these elements are entirely independent. You can use any or all of them selectively as required*

