Agenda

- UVM working group history
- UVM 1.2 plan and key features
- How to contribute to UVM
- Summary and next steps
Formation And Objective

Charter: define standard technology and/or methods to realize a modular, scalable, and reusable generic verification environment

- **Formed February 2008**
  - Co-Chairs: Hillel Miller (Freescale) and Tom Alsop (Intel)

- **Background**
  - IEEE 1800 SystemVerilog lacked standard for VIP creation and use
  - Many methods existed requiring expensive retraining and conversion costs

- **Objective**
  - VIP creation standards to lower verification costs and improve design quality throughout the industry
Key Standard Release Milestones

- VIP Recommended Practices standard and OVM/VMM interoperability reference library
- UVM 1.0 standard and reference library plus user guide
- UVM 1.1 standard and reference library plus user guide (adds register layer)
- UVM 1.2 reference library update release

- UVM 1.0 Early Adopter reference library update release
- UVM 1.1a reference library update release
- UVM 1.1b reference library update release
- UVM 1.1c reference library update release
- UVM 1.1d reference library update release
UVM 1.1 Reference in Widespread Use

- **Actual standard defines UVM APIs**
  - Accellera reference library is commonly identified as “UVM”

- **UVM 1.1 defines features that fulfill charter**
  - Library includes component definition, factory, messaging system, register layer, and more
  - Enables VIP to scale from block to systems
  - Validated on multiple simulators

- **Broadly adopted throughout industry**
  - LinkedIn UVM community has 3700 members
  - UVM LinkedIn membership passed OVM April ‘13
Accellera members and non-members welcome

Active forums provide fast answers from community of experts

Links to technical material, tutorials from Accellera sponsored shows, tool/service/VIP providers and more
Agenda

- UVM working group history
- UVM 1.2 plan and key features
- How to contribute to UVM
- Summary and next steps
UVM 1.2 Improvements

- Standard update required due to API changes
- Backward compatibility remains high priority
- Planned for Q4’13

New features include
- Messaging improvements
- Phasing improvements
- Extensibility of factory and report server
- Command line improvements
- Release of sequence library

Numerous bug fixes
Messaging Improvements

- **Goal:** Provide an object based implementation for UVM and not completely rewrite the API and/or use model

- No changes to existing reporting macros
  - `uvm_info()`, `uvm_warning()`, `uvm_error`, `uvm_fatal` [Note: additional optional args will be added]

- `uvm_report_info/warning/error/fatal()` have additional optional args

- `uvm_report_handler` will be a `uvm_object`

- `uvm_report_server` enhanced to be object-oriented

- `uvm_report_catcher` refactored to communication with `report_server`

- Some methods supporting older infrastructure deprecated but not removed in 1.2
Phasing Improvements

- **Primary phasing architecture of UVM is not in question**
  - Build … connect … run … report are unchanged

- **Focus is on run sub-phasing**
  - Goal: provide standard for time-consuming reset, configuration, run, shutdown

- **Initial run-time phasing implemented in 1.0**
  - No use-model documentation provided
  - Can be challenging for users to implement complex functionality

- **Simplification and documentation planned for UVM 1.2**
  - Project distributed to several “Mantis” items to improve specific functions
  - Use models to be documented in user guide
Extensibility of Factory and Report Server

- Future extensions to UVM depend on a flexible library
  - Bug-fixes and extensions should be non-intrusive
  - Add-on features often require changes to core library
  - All core changes require prolonged study to assess backward compatibility

- Example: static member use
  - Can’t be overridden so changes must be made in the core

- Example: missing encapsulation
  - Class members should have access methods
  - Encapsulation enables traceability for class member access

- UVM 1.2 will enhance extensibility with limited changes to the API
Command Line Improvements and Sequence Library Release

- **Command line processor included in UVM 1.0**
  - Functionality similar to SystemVerilog `$value$plusargs`
  - Several bug-fixes and enhancements address with UVM 1.2

- **Sequence library introduced as beta in UVM 1.0**
  - Replaces string-based sequence library in UVM 1.0 early adopter
  - Extensively tested now
  - Mantis items addressed
  - Will become supported in UVM 1.2 standard
Agenda

- UVM working group history
- UVM 1.2 plan and key features
- How to contribute to UVM
- Summary and next steps
Contributing to the UVM

- Employee of Accellera member companies
  - Contribute your experience in weekly meetings and on reflector topics
  - Code solutions to Mantis items
  - Offer enhancements to the UVM standard and reference implementation

- Everyone everywhere
  - Use UVM!
  - Ask and answer question on UVMWorld forums
  - Contribute code, examples, tips, and more in UVMWorld contributions area
  - Report Mantis items – both bugs and enhancements
Agenda

- UVM working group history
- UVM 1.2 plan and key features
- How to contribute to UVM
- Summary and next steps
Summary and Next Steps

- **UVM is a well established industry standard**
  - Supported by multiple simulator providers
  - Extensive library of VIP from multiple providers
  - Service and training available worldwide

- **Active UVM support community**
  - UVMWorld forums and contributions areas
  - Supported by Accellera member and non-member experts

- **UVM is a living standard**
  - UVM 1.2 in development for delivery in 2013
  - Improving messaging, phasing, and many other features
Discussion and feedback