February 25-28, 2013
DoubleTree, San Jose

accellera SYSTEMS INITIATIVE

Migrating from OVM to UVM
The Definitive Guide

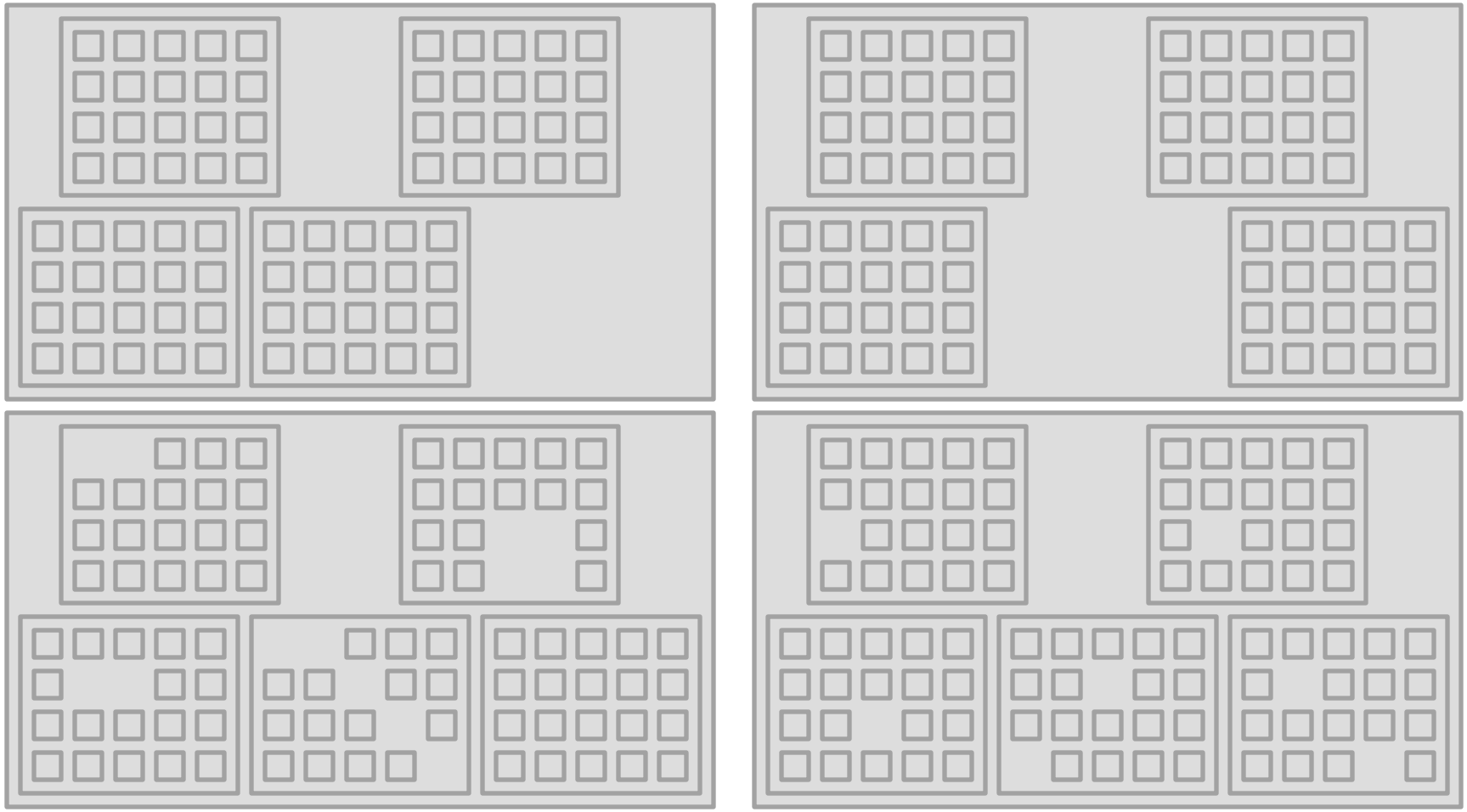
by
Adiel Khan
Senior Staff Engineer
Synopsys

Overview:

- AMD OVM to UVM migration projects
- Prior to OVM 2.1.2
- General UVM base class differences
- OVM to UVM config semantic diffs
- UVM reporting and messaging usage
- UVM new phasing syntax and semantics
- UVM sequence API changes
- Summary

AMD OVM to UVM migration

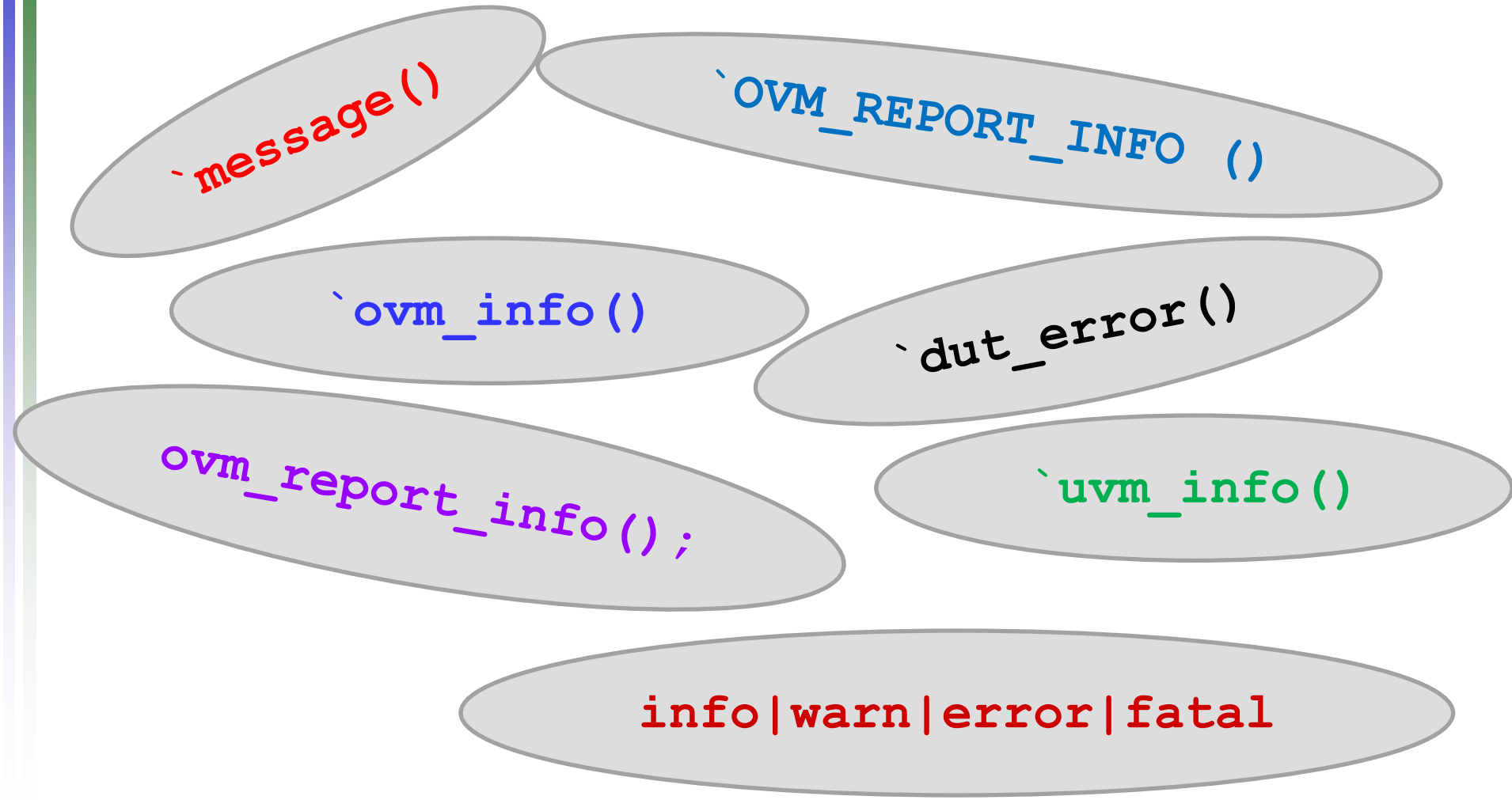
- SoC projects with multi-subsystem and many IPs



Prior to OVM 2.1.2

- Many deprecated issues not highlighted by OVM
- Threaded classes not in later OVM versions
- Reporting API's changed throughout OVM revisions
- OVM simulation termination global_stop_request
- OVM deprecated phases
- Vendor specifics included within OVM library.

Report on the Reporting API



General UVM base class diffs

- Inclusion schemes
- Parameterization improvements
- Field Macro updates
- Comparator fixes

OVM to UVM config semantics

- Mantis 3731 set/get_config semantics
  - OVM implementation
  - OVM truth table
  - UVM 1.0 implementation
  - UVM 1.0 truth table
  - UVM 1.1 implementation
  - UVM 1.1 truth table

OVM set_config/get_config

- set_config_object would clone
  - If the clone bit was set to '1'
- get_config_object would clone:
  - If get_config_object and set_config_object's clone bits were both set to '1'
  - If either were set to '0', then get_config_object would never clone
- apply_config_settings simply used get_config_object
- Note: This is not how it was *documented* in OVM, but it is how it was *implemented*

Truth Table: OVM

Clone bit on			
Set	Get	get_config_object	apply_config_settings
0	0	No clone	No clone
0	1	No clone	No clone
1	0	No clone	No clone
1	1	clone	clone

UVM 1.0 set_config/get_config

- set_config_object would clone
  - If the clone bit was set to '1'
- get_config_object would clone:
  - If the clone bit was set to '1'
- apply_config_settings simply used get_config_object
- Note (again): This is not how it was *documented* in UVM, but it is how it was *implemented*

Truth Table: UVM 1.0

Clone bit on			
Set	Get	get_config_object	apply_config_settings
0	0	No clone	No clone
0	1	**clone**	**clone**
1	0	No clone	No clone
1	1	clone	clone

UVM 1.1 set_config/get_config

- A rather strange compromise
  - An attempt at maintaining backwards compatibility with both OVM *and* UVM 1.0

- set_config_object would clone
  - If the clone bit was set to '1'
- get_config_object would clone:
  - If the clone bit was set to '1'
- apply_config_settings would *not* use get_config_object
  - Instead, it manually implemented the same functionality that OVM provided

Truth Table: UVM 1.1+

Clone bit on			
Set	Get	get_config_object	apply_config_settings
0	0	No clone	No clone
0	1	**clone**	**No clone**
1	0	No clone	No clone
1	1	clone	clone

Other Improvements

- **UVM new phasing syntax and semantics**
  - Basic phasing mods
  - Termination of testcases
- **UVM reporting and messaging usage**
  - Command line control
  - Reporting Classes
  - Printing Methods
- **UVM Sequence API**
  - Debug & Objections
  - Port connectivity
  - Sequencer Relationships & Default Seqs with Libs

Command line message control

- Converting OVM Command line options cause UVM warnings. The format of the example below is an incorrect translation from "O"vm to "U"vm. The verbosity options are reordered under UVM.

i.e:

- `+ovm_set_verbosity=start_of_simulation,ovm_test_top.t0.ioe_s.slave[0].*,OVM_HIGH`

To:

- `+uvm_set_verbosity=start_of_simulation,uvm_test_top.t0.ioe_s.slave[0].*,UVM_HIGH`

becomes :

- `.+uvm_set_verbosity=uvm_test_top.t0.ioe_s.slave[0].*,_ALL_,UVM_HIGH,start_of_simulation`

Summary

- Knowledge of what, how and where modification are required de-risks the migration process from OVM to UVM.
- At AMD and within Synopsys there are technologies and resources that make the OVM to UVM transition painless and can even be done unbeknown to the user.
- For AMD the UVM upgrade is a long term strategy that needs to coincide with project specific needs and requirements.
- The process has been proven and well understood thereby enabling users to take advantage of the OVM bugs addressed by UVM and the new features UVM brings for advanced verification.

THANK YOU FOR LISTENING

- Question and Answer Sessions……..