EE 8350 Advanced Verification Methodologies for VLSI Systems
LAB 3

OBJECTIVE: To learn about the reporting mechanisms in UVM.

UVM has a rich set of message-display commands and methods to change the number and types of messages that are displayed without re-compilation. The same message mechanism also includes the ability to alter the severity of the message in order to adapt to a given verification environment.

The **uvm_report_object** class provides an interface to the UVM reporting facility. Through this interface, components issue the messages that occur during simulation. Users can configure the actions that are taken and the specific messages coming from a particular component, or globally for all messages from all of the components in the environment. Defaults are used when an explicit configuration is not specified.

A report contains an id string, the severity, the verbosity level and the message itself. It can also include the filename and the line number that produced the message.

Format:

```
uvm_report_info("STRING ID","STRING MSG",verbosity);
```

- String ID is used to identify the message.
- Message String contains the message that needs to be reported.
- Verbosity is used to express the importance of the report. If the verbosity level of a report is higher than the specified maximum verbosity level of its report object, then it will be ignored.

Example:
To print the value of variable '**var**':

```
uvm_report_info("INFO",$sformatf("The value of variable 'var'
= %d",var),VERBOSITY);
```

Reporting Macros:

UVM reporting provides the following set of macros for embedding report messages:

- **`uvm_info(string ID, string MSG, verbosity);**
- **`uvm_error(string ID, string MSG);**
- **`uvm_warning(string ID, string MSG);**
- **`uvm_fatal(string ID, string MSG);**

Verbosity Level:

Verbosity level describes how verbose a testbench will be. The default verbosity level is **UVM_MEDIUM**. If this level is used, any messages with **UVM_HIGH** or above are filtered out. The various verbosity levels and their corresponding numerical values are given in the following table:

Verbosity Level	**Value**
UVM_NONE	0
UVM_LOW	100
UVM_MEDIUM	200
UVM_HIGH	300
UVM_FULL	400
UVM_DEBUG	500

Step 1: Run the following commands on the test directory:

```
vcs -Mupdate +v2k -sverilog -timescale=1ns/10ps +incdir+$UVM_HOME/src
$UVM_HOME/src/uvm.sv $UVM_HOME/src/dpi/uvm_dpi.cc -CFLAGS -DVCS top.sv -l
compile.log +vcs+dumpvars+verilog.vpd -debug_all

./simv +UVM_TESTNAME=uvm_test_1
```

The above commands run the UVM test with the default verbosity level of UVM_MEDIUM. Observe the print statements being displayed in the result. Take a screenshot of the result.

There are two ways to change the default verbosity level:

1. To change the verbosity level for a uvm_component (where ‘comp’ is the object name), use the following command.

   ```
   comp.set_report_verbosity_level(UVM_HIGH);
   ```

2. To set the verbosity level for the entire simulation, we can use the command line option:

   ```
   +UVM_VERBOSITY=UVM_HIGH
   ```

Step 2: Run the test for verbosity levels UVM_HIGH and UVM_LOW. Take screenshots of the results in both cases.

```
./simv +UVM_TESTNAME=uvm_test_1 +UVM_VERBOSITY=UVM_HIGH
```

```
./simv +UVM_TESTNAME=uvm_test_1 +UVM_VERBOSITY=UVM_LOW
```

Simulator action:

The associated default simulator action with each of these severity levels is as follows:

Severity	**Default simulator action**
UVM_INFO	UVM_DISPLAY
UVM_ERROR	UVM_DISPLAY | UVM_COUNT
UVM_WARNING	UVM_DISPLAY
UVM_FATAL	UVM_DISPLAY | UVM_EXIT

Descriptions for the different simulator actions are:

Simulator Action	**Description**
UVM_EXIT	Exit from simulation immediately
UVM_COUNT	Increment global error count
UVM_DISPLAY	Display message on console
UVM_LOG	Captures message in named file
UVM_CALL_BACK	Calls Callback method
UVM_NO_ACTION	Do nothing

Simulator action for each severity level can be modified using the following command:

```
set_report_severity_action(Severity, Action)
```

Step 3: Uncomment the following line in ***uvm_reporting.sv.***

```
//set_report_severity_action(UVM_FATAL,UVM_DISPLAY);
```

Step 4: Run the simulation and take a screenshot of the result.

```
vcs -Mupdate +v2k -sverilog -timescale=1ns/10ps +incdir+$UVM_HOME/src
$UVM_HOME/src/uvm.sv $UVM_HOME/src/dpi/uvm_dpi.cc -CFLAGS -DVCS top.sv -l
compile.log +vcs+dumpvars+verilog.vpd -debug_all

./simv +UVM_TESTNAME=uvm_test_1
```

Observe that the simulation doesn't end after UVM_FATAL. The UVM_EXIT severity has been overridden.

Questions:

1. What are the verbosity levels if a UVM test is run with the command line option `+UVM_VERBOSITY=250` ?

   The simulation is run with the following command:

   ```
   ./simv +UVM_TESTNAME=uvm_test_1 +UVM_VERBOSITY=250
   ```

   Take a screenshot of the result.

2. What happens when you run the simulation by uncommenting the following line in ***uvm_test_1.sv*** ?

   ```
   //rpt.set_report_verbosity_level(UVM_MEDIUM);
   ```

   The simulation is run with the following command:

   ```
   vcs -Mupdate +v2k -sverilog -timescale=1ns/10ps
   +incdir+$UVM_HOME/src $UVM_HOME/src/uvm.sv $UVM_HOME/src/dpi/uvm_dpi.cc -
   CFLAGS -DVCS top.sv -l compile.log +vcs+dumpvars+verilog.vpd -debug_all

   ./simv +UVM_TESTNAME=uvm_test_1 +UVM_VERBOSITY=UVM_LOW
   ```

   Take the screenshot of the result.

3. What is the functionality of `set_report_id_verbosity` ?
4. How do you set the verbosity level for a component and all its children?
5. What do the macros `__FILE__ and `__LINE__ indicate?

What is to be turned in?

A report (pdf file) consisting of the answers to the above questions as well as screenshots for the following results:

1. Simulation result for the default verbosity level.
2. Simulation result for the UVM_HIGH verbosity level.
3. Simulation result for the UVM_LOW verbosity level.
4. Simulation result with the modified action for UVM_FATAL.
5. Simulation result for Question 1.
6. Simulation result for Question 2.

References used in the preparation of this lab**:**
http://www.verificationguide.com
http://testbench.in
http://www.asic-world.com
https://verificationacademy.com
https://colorlesscube.com/uvm-guide-for-beginners
https://www.doulos.com
http://www.chipverify.com/uvm