Simplifying UVM messaging – `g2u_display()

SystemVerilog, UVM, UVM1.2

by UnleashingUVM

As part of our goal to making UVM easy-to-use, we keep adding more features to the Go2UVM library. One of the recent additions is “Simplified messaging in UVM”. Intention is to bring the benefits of UVM messaging to all Verilog-aware engineers in the easiest way. Looking back at Verilog’s native ***$display()***, it is the most used debug techniques in simulation based verification. It is easy to use, concise and almost every Verilog aware engineer is used to this tiny feature. However it does have its deficiencies, especially when it comes to debugging someone else’s code and when the project is large. Some of the most common issues with plain old ***$display()*** are:

- Lack of ***$time*** in-built – i.e. since Verilog is mostly dealing with timed models, it is very useful to have the simulation time available with every message. One can add it manually to every ***$display()*** – but that’s work!
- No FILE & LINE number information from ***$display()*** – as SystemVerilog added 2 handy macros ***`__FILE__*** to get the file name, and ***`__LINE__*** macro to retrieve the line number. This has been a standard debug technique in software for ages especially when the code is expected to change hands often.

UVM has added several APIs and macros to resolve the above issues and the popular macros are:

- ***`uvm_info (ID, MSG, VERBOSITY)***
- ***`uvm_error (ID, MSG)***

However, a uninitiated Verilog like engineer finds the above bit strange, especially the ID part of the macros. With no strict guidelines from UVM documentation on what to use as ID, teams tend to misuse this quite a bit. Seasoned engineers tend to use built-in ***uvm_object::get_name()*** as ID as that tends to make the log file comprehendible and correlates quickly to the objects in UVM. And the VERBOSITY – many a times engineers use UVM_MEDIUM. Given that SystemVerilog allows default values for macro arguments, one would hope and expect that the UVM library would add a default value for this argument. – but as of now it does NOT – forcing users to add this to every display message by hand! Let’s look at few sample UVM message code:

```
`uvm_info (get_name(), “Hello World”, UVM_MEDIUM)

`uvm_info (get_name(), “Generated a packet”, UVM_HIGH)

`uvm_info (get_name(), $sformatf (“Writing to addr: 0x%x data: 0x%x”,

x0.addr, x0.data), UVM_MEDIUM)
```

That’s bit of a code for first time UVM engineers. How about we simplify the same as below:

```
`g2u_display (“Hello World!”)

`g2u_display (“Generated a packet”, UVM_HIGH)

`g2u_display ($sformatf (“Writing to addr: 0x%x data: 0x%x”,

x0.addr, x0.data))
```

That is handy and neat! Under-the-hood we still call the same UVM API (***uvm_report_info()*** for instance). So what happens?

1. ID : we use ***get_name()*** automatically (And a nice, built-in workaround to handle scopes outside ***uvm_object***, don’t worry!)
2. VERBOSITY – we use default argument to the macro to use UVM_MEDIUM

So with Go2UVM, adopting UVM becomes easier and simple.

Stay tuned for more updates in next major release of ***Go2UVM***!