Novel GUI Based UVM Test Bench Template Builder

Vignesh Manoharan
Aeva Inc.
555 Ellis St. Mountain View, CA 94043

Abstract

Adoption rate of Universal Verification Methodology (UVM) is increasing day by day across industries and the need for building new Verification Intellectual Property (VIP) or testbench is in great demand. Writing effective and structured UVM testbench from scratch is cumbersome most of the time and following a standard structure with provision for better re-usability across projects is also challenging. What if the time taken for initial development cycle is reduced to minutes instead of days with the help of a Graphic User Interface (GUI) to build the verification component templates? This poster presents an overview about the GUI interface used to develop the individual UVM components or the entire VIP templates loaded with features to customize and configure as per the user requirements.

Deep Dive into UVM Template Generator Operation

The tool is built using Python framework to create the GUI layouts in grid-based mechanism. All text processing and editing are done using python scripting. The tool helps in:

- Building pure UVM template code
- Building single UVM components or complete UVM testbench and architecture
- Building Multi Agent, Multi Monitor, Multi scoreboard based Environments
- Building Multi-Environments based flow targeting complex SOC's (System on Chip) scenarios
- Integrating Agents, Monitors, Scoreboards into already existing Environment and helps in integration between environments
- All the codes generated from this tool uses Natural Docs' formatting for easier documentation

Creating Single UVM Components

The moment the user launches the tool, the GUI pops up with two options, namely 1. create “Single UVM Component”, 2. “Single & Multi Env VIP” as shown in Figure 1.

Once the user clicks the single component radio button, tool lists out multiple objects and component options to create namely: sequence_item, agent, environment etc. as shown in Figure 2. The user can choose whichever component or object they want to create and the corresponding template by clicking the ‘Generate’ button. Based on the component the user chooses, the tool displays required customization options.

Interface Creation: The tool provides the user with multiple options to develop an interface file such as:
- Creating a default interface with an empty shell
- A user defined interface via the GUI as shown in Figure 3.
- Loading a spreadsheet

Agent Creation: When the user wants to create an agent, the tool further provides options for the user to enter the number of driver-sequencers or monitors they want with required names as shown in Figure 4. The tool generates the necessary code templates which are compile clean and ready to use. As soon as the code is generated, the tool goes to the default/initual layout.

Once the user has provided the required details about all the environments and the monitor-scoreboard connectivity information, the user needs to click the “Done Env Cfg” button as shown in Figure 9 to instruct the tool that the user has confirmed all testbench setup and it is safe to move ahead.

After confirming the environment configuration, the user then clicks the “Generate Code” button as shown in Figure 10. This will instruct the tool to build all testbench codes, necessary files, and directory structures.

In the complete UVM VIP tool window, when the user chooses the “Load Spreadsheet Approach”, the tool pops up with the layout as shown in Figure 11.

In this mode, the user can enter the details in a tool understandable spreadsheet format as shown in the Figure 12.

Creating Complete UVM VIP

The moment user clicks the “Single & Multi Env VIP” from the initial layout, the tool provides a couple of options as shown in the Figure 5, namely
- 1. GUI Approach
- 2. Load Spreadsheet Approach
The complete GUI based approach tool layout is shown in Figure 6.

As the user starts filling in the details about the environment to be created, tool intuitively brings up the required widgets to provide necessary details. For example, when the user starts filling the details about the agent, the tool provides input widgets to enter the details about driver, agent level monitor and the interface information as shown in Figure 7.

Next, the user enters the all required information about the environment to be built, the user will click the ‘Env Setup’ button found at the bottom of the tool window. By clicking that button, the tool generates a matrix table with all the monitors, scoreboard, and provide the option for user to make the necessary connection as shown in Figure 8.

The tool knows the user is adding new components onto an existing block level environment, thus the tool generates only the new components and then stitches them onto the already existing environment in all the necessary places.

The granularity at which the work can start from can be:
- Adding single or multiple port connectivity's between monitor and scoreboard
- Adding new environment level monitor’s and scoreboard’s in already existing environment with updated connectivity
- Adding driver/monitor inside already existing agents and adding new agents into existing environment
- Adding an environment using create & stitch or just stitch process into an already existing environment

Sample Code Snapshots

In Figure 13, the user creates a single component named “Vignesh” and adds a new environment, the driver/monitor inside the existing agent. Additionally, the user creates a new environment, namely “Vignesh1”.

Novel Stitch, Create & Stitch feature

The art of developing a testbench doesn’t happen in a single day but is a continual long-term process. For example, on day 1, the user might just need to build the environment skeleton. On day 2, the user might end up adding few other components namely agents and environment level monitors. Later the user adds the required scoreboard and connectivity. How does this tool take care of such cases? Well, the tool provides couple of novel features namely, “Stitch”, “Create & Stitch” modes which helps in incremental testbench development process.

Stitch Mode: This mode comes in handy if the user has already created an environment with the tool which takes care of generating different kinds of clock sources and now the user wants to build a block level benchmark which is going to take care of register holpling. The user needs to launch the tool, generate the required block level testbench skeleton and then using the “Stitch mode”, the user can stitch the other sub-environments into this block level environment. The user needs to enter the required number of sub-env’s wanted to be stitched and add the sub-env’s names appended with “_<n>” as shown in the Figure 14.

Summary

The UVM template generator provides the user to create any component template or the entire VIP dynamically in matter of minutes. The generator helps in standardization of code development and re-usability of the code across the projects and helps in complete integration of the verification collateral. With the template generator’s unique ‘Create & Stitch’ feature, the tool can add new components, add connection between components or append sub environment VIPs to the already existing code and help in incremental enhancement of the testbench. Hence, this tool indeed improved verification productivity and showcased its performance in complex streamlined products.