Real-Time Synchronization of C model with UVM Testbench

Kirtan Mehta

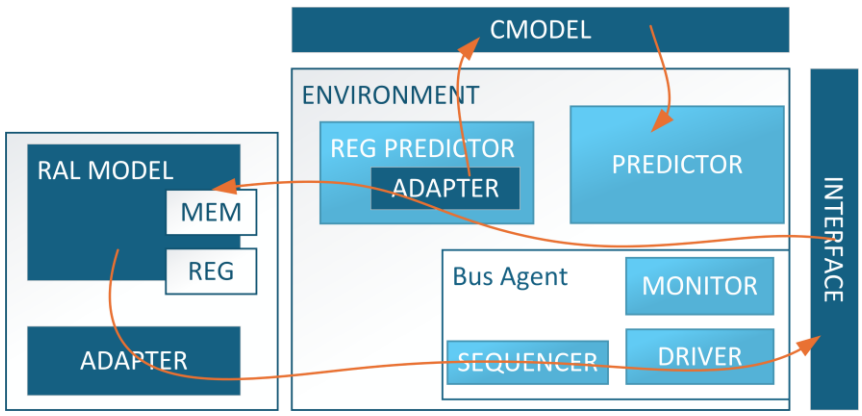
onsemi

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

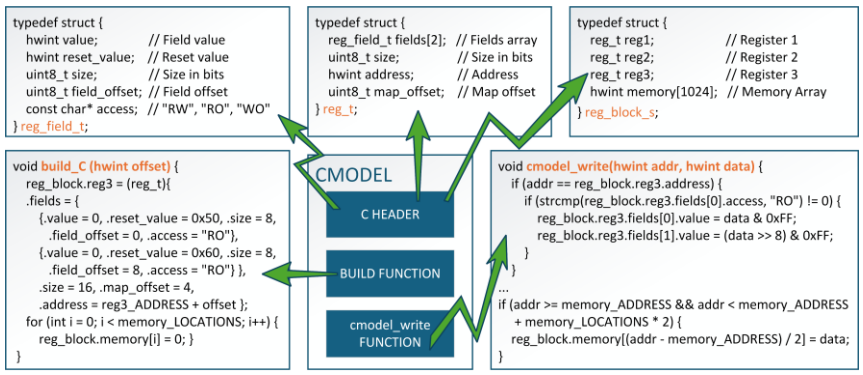
INTRODUCTION

- In ASIC verification, C-based models are crucial for handling intricate computations and serving as golden reference models, streamlining testbench development and Post-Silicon Validation.
- However, integrating these Cmodels with UVM Testbenches which are often implemented in Scoreboards and Predictors, presents challenges in synchronization with RTL designs, requires manual intervention, code development, often leading to inefficiencies and potential errors.
- By embedding DPI-C functions into UVM Register Abstraction Layer (RAL) adapters, we enable real-time monitoring and dynamic updates of bus traffic.
- This approach not only reduces the need for manual synchronization but also enhances the accuracy and efficiency of the verification process.
- Automating synchronization through DPI-C significantly reduces verification time and minimizes the risk of manual errors, enhancing the overall efficiency and reliability of ASIC verification.

UVM TB ARCHITECTURE WITH CMODEL CONNECTED TO RAL ADAPTER



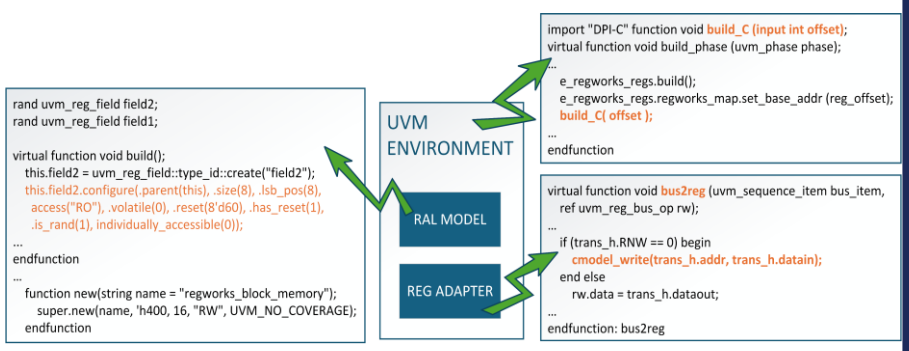
C-MODEL IMPLEMENTATION



- Register, reg fields and reg block structures which could be reused in C to mimic RAL and Design.
- Build function initializes the Cmodel registers and memories by allocating space and defaulting the values.
- Cmodel write function takes in Address and Data as input and based on Register configuration/access, address and updates the data.

UVM TB IMPLEMENTATION

- The image below depicts various components, such as the UVM Environment, RAL model, and REG Adapter, where DPI functions are implemented.
- These functions ensure synchronization between the C-Models, RAL, and the UVM Testbench.



IMPLEMENTATION

- The UVM code snippets defines and configures registers, register fields and memory using RAL class, while the C code snippet mirrors this by defining and initializing C structures with its register map and register fields, ensuring identical configuration and initialization.
- The function build_C constructs the C model with specified register fields and initializes memory locations. This occurs at the same time as RAL ensuring the TB, Design and Cmodels are in sync.
- The cmodel_write function writes data to specific hardware addresses in a simulation environment by updating register fields or memory locations based on the provided address and data.
- This ensures accurate simulation of register operations and interactions between the processor and peripheral devices.

ADVANTAGES

- **Automation and Efficiency:** Embedding DPI-C functions into UVM RAL adapters reduces manual effort and potential human error.
- **Real-time Updates And Monitoring:** Ensures continuous updates and synchronization between the C model and testbench for accurate and efficient verification.
- **Scalability:** Handles complex operations like burst via front-door memory transactions, suitable for block-level and SoC verification environments.
- **Consistency:** Mirrors the register and memory structure of the UVM RAL model, enhancing uniformity and reliability across the codebase.

DRAWBACKS

- **Complex Integration:** Integrating DPI-C functions into RAL adapters requires common definitions. Continuous synchronization with evolving designs adds complexity.
- **Performance Issues**: Real-time updates may slowdown TB simulation time; Larger TB might require additional support to handle this automation.

AUTHOR INFO

KIRTAN MEHTA, Contact info: kirtan.mehta@onsemi.com