UVM 入门和进阶实验 5

嗨，我们终于携手走到了 UVM 入门的最后一个实验，关于 UVM 寄存器模块的应用。通过对 UVM 寄存器模块的学习，我们将会在本次实验中掌握：

- 对 uvm_reg 的定义，以及 uvm_reg_block 的组织。
- 对 uvm_reg_adapter 的定义，以及它与 uvm_reg_block 之间的关系。
- 对 uvm_reg_predictor 的使用，以及它与 uvm_reg_adapter 和 uvm_reg_block 之间的关系。
- 改造之前的寄存器发送序列，并以 uvm_reg 的操作方式去取代。
- 应用内建的寄存器序列， 做全面的寄存器测试。

寄存器模型的完善和嵌入

在接下来的实验中，同学们可以看到，uvm_reg 和 uvm_reg_block 的定义已经完成。请结合红宝书 14.1 节，来理解 mcdf_rgm 寄存器模块的定义。同时，请继续完成以下的实验步骤：

- 实验 1.1， **请实现 reg2mcdf_adapter 类的方法 reg2bus 以及 bus2reg**，可参考红宝书 14.2.3 节。
- 实验 1.2，请在 mcdf_env 中分别声明 register_block，adapter 和 predictor，并完成例化。同时，在 connect 阶段中，请组织它们的关系，做必要的句柄连接。

寄存器模型的使用

寄存器模型很大的一个优势在于它的抽象性和复用性，接下来，我们将会改造之前的激励序列。请按照以下要求完成实验：

- 实验 2.1，请完成 register block 句柄的传递。
- 实验 2.2，请将 mcdf_data_consistence_basic_virtual_sequence **原有的由总线 sequence 实现的寄存器读写，改为由寄存器模型操作的寄存器读写方式**，可参考红宝书 14.2.5 节。
- 实验 2.3，请将 mcdf_full_random_virtual_sequence 原有的由总线 sequence 实现的寄存器读写，改为由寄存器模型预先设置寄存器值，再统一做总线寄存器更新的方式，并且稍后由后门读取的方式取得寄存器值，加以比较。

寄存器内建序列的应用

参考红宝书 14.3.5 节，UVM 已经内建了一些寄存器序列，在接下来的实验中，我们将选择一些序列对寄存器展开全面测试：

- 实验 3.1， 在新建的 mcdf_reg_builtin_virtual_sequence 类中，请使用 uvm_reg_hw_reset_seq， uvm_reg_bit_bash_seq 和 uvm_reg_access_seq 对 MCDF 寄存器模块展开全面测试。在仿真过程中，请注意理解寄存器测试序列中打印的消息，对照红宝书表 14.6，理解这些内建寄存器测试序列的作用。

经过六次 UVM 入门和进阶实验的"艰苦作战”，UVM 新兵训练终于可以结束了！恭喜你，已经成为了一名新鲜出道的 UVM verifier。 在短暂的兴奋过后，路桑还会将你送上“真枪实弹” 的 UVM 实战现场，去零距离地感受一什么是真实的硬件设计 (MCDF 在设计上的升级版本)，以及什么是更加完善的验证环境(现有 MCDF 验证环境的升级)。在接下来的 UVM 实战模块中，我们将进一步降低授课时间而提高编程时间， 目的仍然在于培养大家在"眼到”之后，可以“手到”和“心到”，知行合一。那么，就让我们在接下来的 UVM 实战模块中见吧！