

谈谈开发语言：从 Verilog 到 SpinalHDL

如题，本期我们不谈模拟，不谈软错误，也不谈电机算法，我们谈谈开发语言。在数字电路设计领域，特别是超复杂的集成电路设计中，存在一匹极其“不讲道理”的黑马——Spinal HDL。SpinalHDL 是啥？这是一种高级语言，跟另外一个高级语言 chisel 相比，确实名气稍微欠缺一些。SpinalHDL 和 Chisel 师承一脉，都是基于 Scala 来进行电路描述。其本质上仍旧是 HDL 而非 HLS，在设计之前依旧需要有清晰的电路结构，只不过是在电路描述上更加方便和快捷而已。那么为何我们要谈 Spinal 呢？这就要从 Verilog 说起了。

历经 41 年风雨洗礼的 Verilog，早已稳坐 FPGA 与 ASIC 开发的主流硬件描述语言宝座，其地位之稳固，自然毋庸置疑。由于拥有与 C 语言相似的语法结构，Verilog 的学习曲线相对平缓，代码清晰易读，语法简洁易懂。

然而，随着芯片规模的不断扩大和复杂度的日益增加，这个历史悠久的语言也逐渐暴露出其局限性。冗长的代码描述使得从逻辑设计到 Verilog 代码实现的过程变得耗时费力，而过于简单的语法又往往让工程师陷入重复且无趣的编程任务中。

即便费尽九牛二虎之力完成了 Verilog 代码的编写，其薄弱的代码检查功能也常导致一些低级错误逃过检查，使得工程师不得不反复返工，这些错误会在仿真，甚至在综合时才能暴露出来。而基于 Verilog 扩展的 SV (SystemVerilog) 也未能从根本上解决这些问题。在仿真验证应用上，SV 虽较 Verilog 有所进步，但在可综合逻辑上并无显著变化。

具体而言，Verilog/SV 存在以下难以克服的痛点：

- 代码重用性差：对于相同用途的逻辑，很难用同一个 function 去调用，除非两边完全相同；否则，即便只是数据的位宽或格式稍有变化，也无法使用。
- 接口连接不便 / 例化模块困难：例化一个模块时，每个接口、每根线都必须逐一连接。例如，一个 AXI4 总线就需要至少 20 次独立连接，这在系统性的连接时简直是灾难性的。

- 缺乏现代化的开发工具：在编写时，没有工具能实现基于变量类型的自动联想、在线的基础语法检查。
- 查错功能薄弱：编译时几乎没有基础的语法检查，导致很多基础问题（如位宽、接口方向、Latch、多驱动、Loop 等）会被带到仿真乃至综合阶段。

这些问题会显著延长芯片的开发周期，甚至可能影响芯片流片的最终效果。

那么，是否有更可靠的解决方案呢？

答案就是我们前文所说的，目前最主流的方案 —— 基于高级语言扩展的 HDL，其中应用最多的是 Scala-Base 的方案，包括 Chisel 和 SpinalHDL 两个成员。

Chisel 因其出身高贵而享有极高的知名度，但对于习惯了 Verilog 的用户来说，其使用和学习方式可能过于“软件化”，参照 Chisel 的标志性项目“Rocket”来看，整个代码充斥着隐式传参，时钟域也被隐式代入，这容易让“老逻辑人”感到不适。大多数想尝试 scala 开发的“选手”，在经历了 chisel 的难度后无不感觉心力交瘁，甚至放弃。

相比之下，SpinalHDL 则更合传统“硅农”的口味。其结构使得即使是零基础的用户也能轻松上手，无需精通 Scala 即可开始编写代码。而且，其强大的查错能力和参数化能力可以显著提高代码质量和可配置性。

SpinalHDL 的优势包括：

- 现代化的 IDE 支持：提供基于类型的代码联想、在线查错、跨模块调用等功能，甚至支持最新的 AI 工具，极大地提升了开发效率。
- 强大的查错能力：在代码生成阶段就能查出诸如位宽匹配、接口方向、多驱动、无驱动、Latch、Loop、跨时钟等问题，避免了在综合阶段才发现的错误。
- 强大的参数化能力：参数可以是数字、算法甚至一段逻辑或一个模块，极大地增强了模块的可配置性。

- 基于 Scala 高级语言的函数调用：支持多参数、多返回值函数调用，显著提高了代码复用度。
- 接口连接方便：接口可以通过例化整体连接，无需考虑方向；也可以通过名字直接连接，无需定义接口。
- 生成代码可读性高：转出的 Verilog 代码基本可以与 Spinal 代码一一对应，方便后续的仿真和前后端工作。
- 无逻辑损耗：作为 HDL，SpinalHDL 在转化过程中没有逻辑损耗，不会产生冗余逻辑。

应用 SpinalHDL 在实际开发时，能够将工程师从琐碎的查错中解放出来，减少人力的使用，并大幅缩短设计开发周期，对于整体工程的开发大有裨益。在面对复杂和繁琐的 FPGA 或 ASIC 开发时，不妨尝试一下 SpinalHDL，它或许能给你带来意想不到的惊喜和收获。

最后补充一点，无论是采用 chisel 还是 SpinalHDL 开发，两种语言都是解放双手的优秀开发工具，优秀的 scala 开发者的开发效率甚至能够抵得上 3 - 4 个传统 verilog 开发者，这是多数公司在实际经历后得出的结论（并非夸张）。无非对于同样都是基于 scala 开发的高级语言来说，SpinalHDL 显得更为符合“工程化”开发，易于上手，而 chisel 更适合学术研究罢了。