

# 覆盖率

覆盖率是用来描述验证完备性的词汇。

## 覆盖率的类型

分为代码覆盖率与功能覆盖率。

- 代码覆盖率
  - 行覆盖率：多少行代码已经被执行过。
  - 路径覆盖率：在穿过代码和表达式的路径中有哪些已经被执行过。(branch if else)
  - 翻转覆盖率：哪些单位比特变量的值为0或1。
  - 状态机覆盖率：状态机哪些状态和状态转换已经被访问过。

代码覆盖率用于分析最终设计执行了多少代码。不添加任何额外的HDL代码，工具会通过分析源代码和增加隐藏代码来自动完成代码覆盖率的统计。当运行完所有测试，代码覆盖率工具便会创建相应的数据库。仿真器都带有代码覆盖率的工具，覆盖率数据也可被转换为可读格式。

- 断言覆盖率

有些断言可以用于查找感兴趣的信号值或者设计状态。可以使用 cover property来测量这些关心的信号值或者状态是否发生。在仿真结束时，仿真工具可以自动生成断言覆盖率数据。断言覆盖率数据以及其它覆盖率数据都会被集成在同一个覆盖率数据库中，verifier可以对其展开分析。

- 功能覆盖率
  - 用于衡量设计在实际环境中的行为是否正确
  - 与设计意图紧密相连接(feature list)

## 功能覆盖策略

在写测试代码之前需要清楚设计的关键特性，边界情况，以及可能的故障模式。

- 收集信息而非数据
- 测试关心的内容
- 测试的完备性

在开始项目时，代码覆盖率和功能覆盖率都很低。接着你开始测试，并且使用不同的随机种子反复进行到功能覆盖率不再增加。这时，创建额外的约束和测试去开发新的区域。保存那些给出高覆盖率的测试和种子组合，以备回归测试之用。

## 覆盖组详解

覆盖组与类相似——一次定义后可多次例化。它包含有覆盖点、选项、形式参数和可选触发（trigger）。一个覆盖组包含了一个或多个数据点，全都在同一时间采集。覆盖组可以在程序、模块或类里定义。在所有情况下，覆盖组都要进行明确的实例化才可以开始采样。