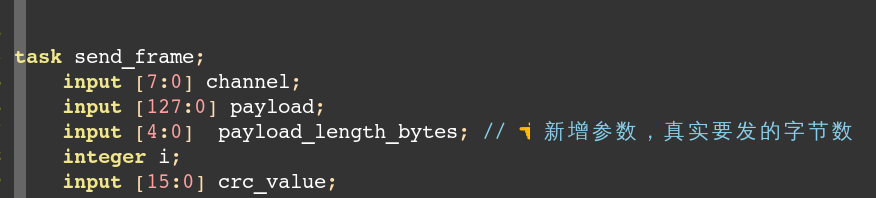
一、自动化任务

在设计代码编写好后，编写testbench进行测试。在testbench内实例化被测top模块，并连接相应信号。在testbench中编写帧序列发送任务：

输入；1. 128bit数据

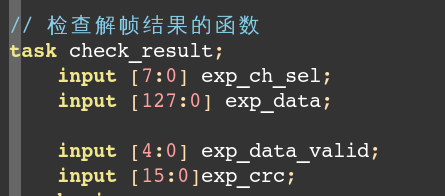
1. 数据有效字节数
2. 16bit CRC
3. 8bit独热码通道选择信号



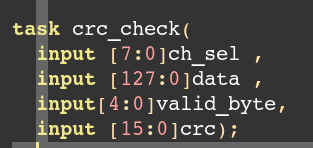
该任务将上述输入组合成一个完整的帧，并依据spec时序要求，依次发送帧头、通道选择、数据、CRC、帧尾等部分。

编写4个子模块的输出检查任务。每个任务会检测对应子模块的输出并与理想输出值比较，并根据比较结果输出相应的打印信息。

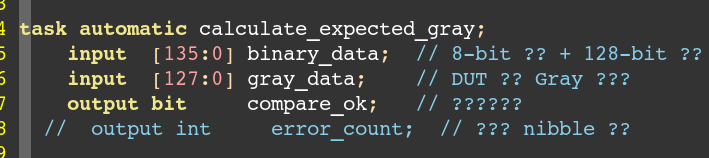
1. 解帧结果检查任务



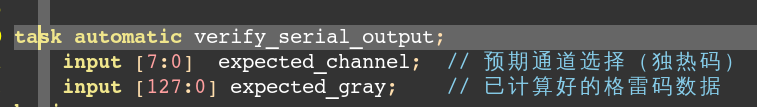
1. CRC校验输出检查任务：



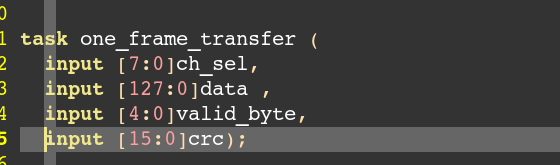
1. 格雷码转换检查任务：



1. 串行输出检查任务：



最后将上述任务封装成完整的帧序列发送+结果检查任务：



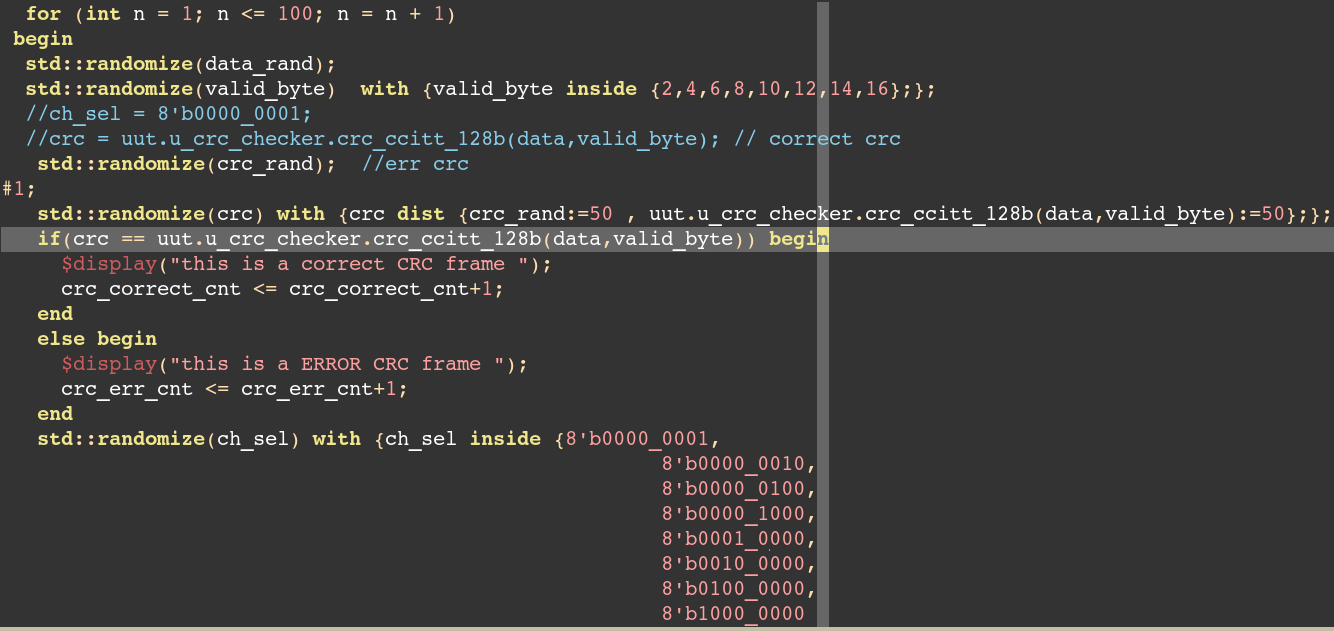
在testbench中对DUT发送一组激励时，只需要调用one\_frame\_transfer任务，testbench就会根据任务输入的data、CRC等数据自动发送相应帧，并检测DUT输出是否满足要求。

1. 测试用例构成

对DUT的测试包含随机测试+定向测试两大类。随机测试用于发送data、CRC等信息随机的帧，而定向测试用于测试边界条件帧、特殊帧等随机测试难以测到的情况，以及补充覆盖率。

1. 随机测试

格式正确的随机帧：



格式正确的随机帧包含符合spec要求的帧头、帧尾以及各个字段，并且各个字段的位宽均符合要求。包括：

1. data：数据为任意随机数，位宽为{16，32，48，64，80，96，112，128}间的随机值。
2. CRC：数据为正确的CRC或任意随机数（错误的CRC），各50%概率。位宽为要求的16位。
3. ch\_sel：数据为随机独热码，位宽为要求的8位。

在所有帧字段随机化后，可以调用one\_frame\_transfer任务发送帧并检查结果。通过for循环或repeat循环，可以进行批量随机测试。

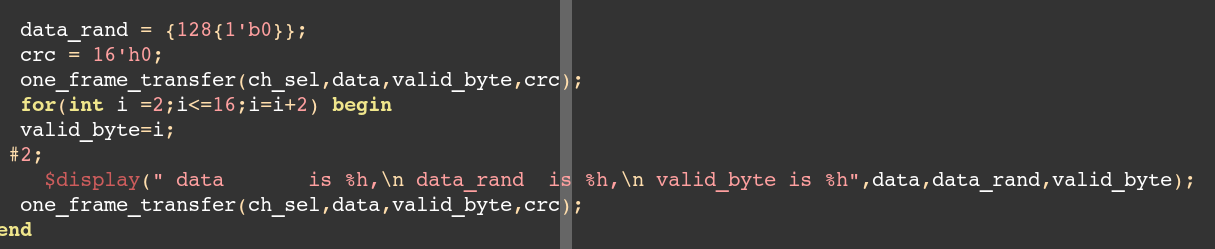
格式错误的随机帧：

错误帧的各字段位宽不完全符合spec要求。格式错误包括帧头错误、帧尾错误、各字段位宽错误、ch\_sel非独热码等错误。通过发送具有不同具体错误的随机帧，验证DUT是否能正确处理错误帧。

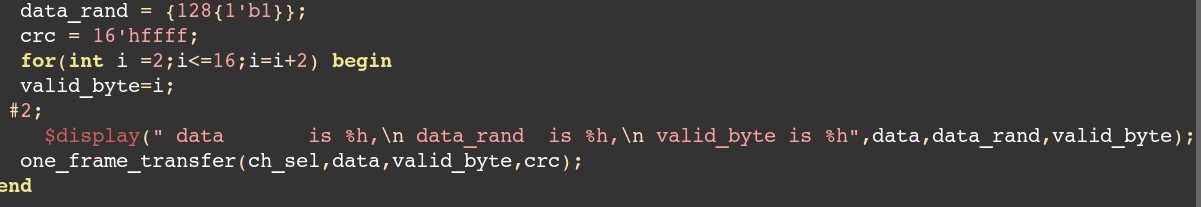
1. 定向测试：

定向测试主要用于边界情况测试、特殊情况测试以及覆盖率的补充。

数据全0测试：

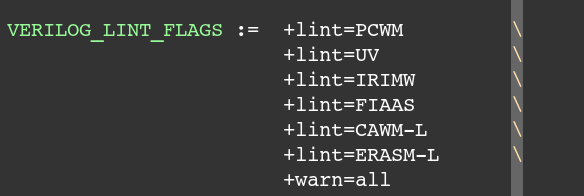


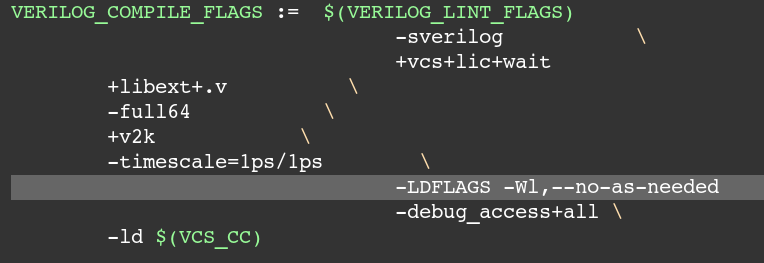
数据全1测试：

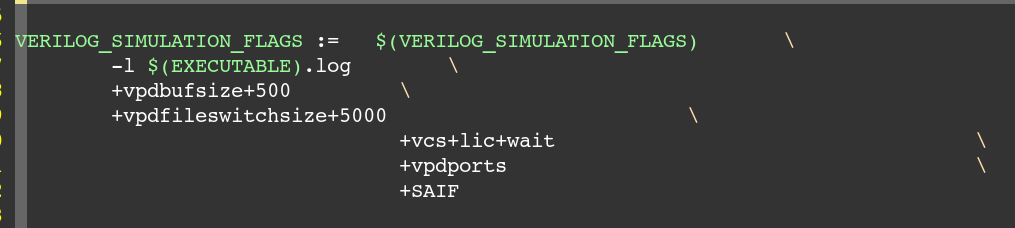


1. 编译仿真命令

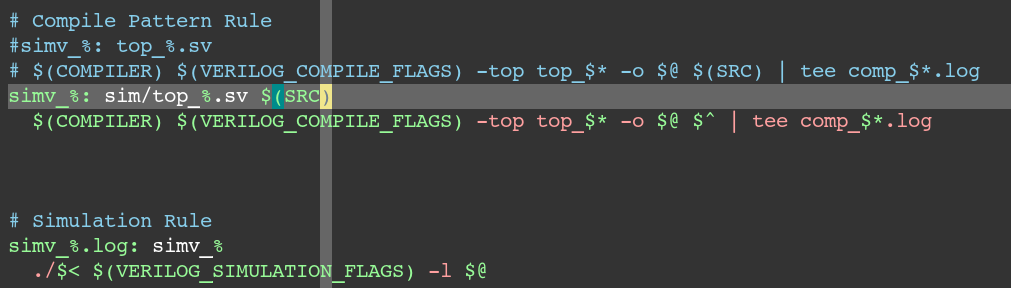
编写makefile脚本来进行自动化编译仿真。Vcs编译仿真选项：







Vcs编译加仿真指令：make simv\_tb.log



清理编译文件命令： make clean

