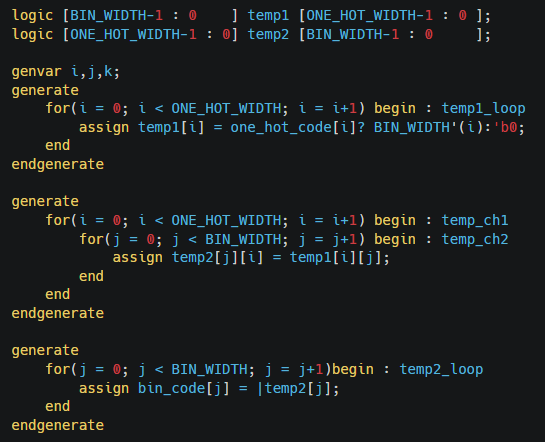
# 部分代码实现思路解析

## onehot\_to\_bin

### 1.1 实现思路

在backup目录下有一版实现，在common\_cells目录下有另一版实现，两者的实现思路实际上是一致的。

backup目录下实现的思路更为直接，关键代码如下所示：



onehot\_to\_bin关键代码

核心过程分为两个步骤：

步骤1：

遍历输入的独热码的所有位，将值为1 的下标位置记录下来 (下标从0开始，实际上这个下标位置就是转化后的二进制码)。记录的方式: 采用一个数组，数组的深度为独热码的位数，数组的每个元素的位宽为独热码的位宽，将独热码元素为1的索引，对应的数组元素赋值为该位置的索引，其他的元素都赋值为0。

步骤2：

经过步骤1后，得到了一个数组，其元素只有一个非0元素，其他值都为0。问题转化为，在这样一个数组中，将这个非0的元素取出来即可。采用的方式为将所有的元素按照对应的位进行或运算（ 这种思路在数字中很常用）。

onehot\_to\_bin的实现思路示意图如下图所示：



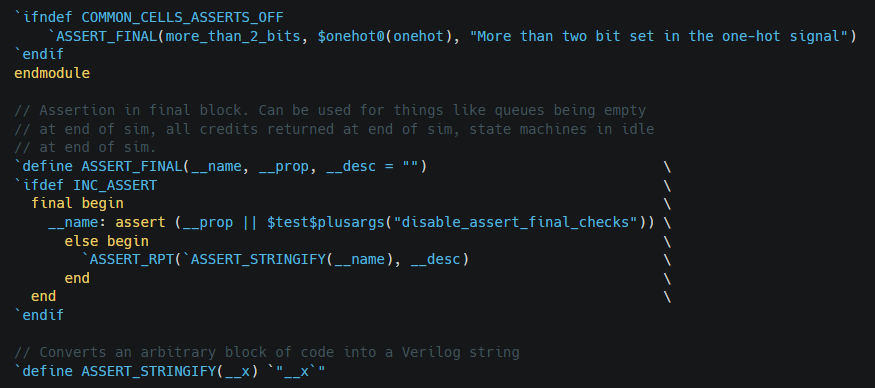
onehot\_to\_bin实现思路

backup下的代码就是如上所述，先在数组中记录索引信息，然后再从数组中将唯一的非0索引取出。

common\_cells下的代码实际上从二进制码的最低位到最高位依次考虑,如上图所示，最后只会有唯一的1对应的那列数据被保留下来，也就是最终的二进制码。

两种实现实际上只是步骤1和步骤2的先后顺序变化。(可以学习这2种实现方式的考虑思路)。

### 1.2 断言相关内容记录



断言相关代码

final begin … end 是一个Verilog块，表示断言在仿真结束的时候进行检查。final关键字确保断言在仿真周期的最后阶段进行评估。

assert( \_prop || $test$plusargs("disable\_assert\_final\_checks”)) else begin

…

end

$test$plusargs("disable\_assert\_final\_checks")：这是一个系统任务，检查仿真命令行中是否包含 +disable\_assert\_final\_checks 参数。如果包含该参数，断言检查将被跳过。

Verilog final关键字:

final关键字是用于指定某段代码或者逻辑在仿真结束时执行，是仿真时的一个特殊块，可以用于做一些最终的检查、报告或者其他需要在仿真结束时执行的操作。

final 关键词的作用：

1. final关键字用于标记代码块，这些代码将在仿真结束时执行。仿真结束时，final块中的逻辑会被执行一次，通常用于报告、检查或者清理工作。
2. final语句通常用于做最后的检查或者错误检测，在设计中非常有用，尤其是验证模型、断言检查等。

仿真完成后，所有final块会被执行一次。

final关键字通常用于执行一些验证性的工作，比如检查信号的状态、断言或者生成测试报告。

该块内的代码只有在仿真完成时才会执行，对于时序信号,final确保了在仿真结束时检查设计的最终状态。

final关键字和断言(assert)常常一起使用，特别是希望在仿真结束时验证某些重要的设计条件时。

final确保这些断言仅在仿真结束时执行，从而避免了仿真过程中出现不必要的错误触发。

final使用的两种形式：

形式1：

final begin

assert(condition) else

$fatal(“error: condition failed.”);

end

形式2：

final assert( condition ) else

$fatal(“error: condition failed.”);