

数字电路高层次综合设计 第三周作业

范云潜 18373486

微电子学院 184111 班

日期：2020 年 9 月 28 日

1 设计与测试综述

本节对实验的设计方法和测试方法进行说明，在具体模块的实现后不再赘述。对单个模块进行系统层次的介绍，即输入、输出以及数据关系的说明。

1.1 设计方法

对于结构级设计方法或者说从门级进行设计的方法，产生的硬件有着最高的确定性，但是设计较困难。在本实验设计的简单逻辑中仍可接受，若是与复杂计算逻辑相关，本这样的方法很难迅速的实现，并且最终结果也缺乏可读性，难以修改。

对于数据流式的设计方法，设计思路直观，设计迅速，但是底层实现不能确定，和工艺以及目标平台相关。这是在较高层次的设计中比较常用的方式。

在本次实验中仅涉及到组合逻辑，我们对于输入的下一节点进行分组，可以得到不同节点的计算式，重复应用直到输出级即可完成设计。

1.2 测试方法

对于所有的模块的 TestBench 均采用 Golden Model 方式进行测试，通过 Python 生成多组随机数据作为输入，计算后得到输出，将输入输出写入文件。在 TestBench 中通过 readmemh/readmemb 读取到测试 Buf 中，逐个输入后比对输出，若是输出错误将信息 display 到标准输出便于调试。同时，将所有的变量的波形进行存储，便于调试。需要注意的是，测试应该使用逻辑全等也就是 `!==`，`===` 来进行测试，防止悬空引脚的高阻态全等。

对于同一模块不同实现可以使用同样的，通过 `'define`，`'ifdef` 等实例化不同的模块。

对大型的工程应该使用 makefile 等方式对测试单元进行管理，如引用库等，但是本次实验仅涉及到简单的逻辑模块调用，无需如此管理。

2 7458 芯片

系统输入：两组三位输入与两组二位输入；系统输出：两组单位输出；系统功能：对两组相同位宽的输入进行与操作，再将两组的结果进行或操作，分别输出。

按照芯片的门级表示进行门级元件调用（结构级）或者通过 `assign` 或 `always`（数据流）进行两种方式的实现。

3 向量逆序输出

系统输入：一个 32 bit 或 4 byte 的数据；系统输出：一个 32 bit 或 4 byte 的数据；系统功能：对 byte 顺序进行倒置，如 {b1, b2, b3, b4} 会引起 {b4, b3, b2, b1} 的输出（每一个 b* 都代表一个 8 bit 数据）。

通过 Verilog 的位选语法进行分组与重组。

4 加法器

系统输入：两个加数，一个进位；系统输出：一个与加数同样位宽的和，一个进位；系统功能：对两个加数和进位进行加和，将产生的和的主体与进位输出。

对于 16 位加法器的两个加数和一个进位通过 `assign` 完成数据加法，输出到 `cout, sum`

对于 32 位的加法器，需要由 16 位的加法器进行串联，第一级的进位为第二级的输入，进行模块调用即可。