

实验一 多数表决器的设计与实现

一．实验目的

1. 掌握基于 Vivado 的数字逻辑电路设计流程；
2. 熟练使用 SystemVerilog HDL 的行为建模方法对组合逻辑电路进行描述；
3. 熟练使用 SystemVerilog HDL 的结构建模方法对组合逻辑电路进行描述；
4. 掌握基于远程 FPGA 硬件云平台对数字逻辑电路进行功能验证的流程。

二．实验环境

1. 操作系统：Windows 10 或 Ubuntu 16.04
2. 开发环境：Xilinx Vivado 2018.2
3. 硬件平台：远程 FPGA 硬件云平台

三．实验原理

1. 3-8 译码器 —— 74LS138

74LS138 是 74 系列集成电路芯片中的一种，实现了 3:8 译码器的功能，如图 1-1 所示。

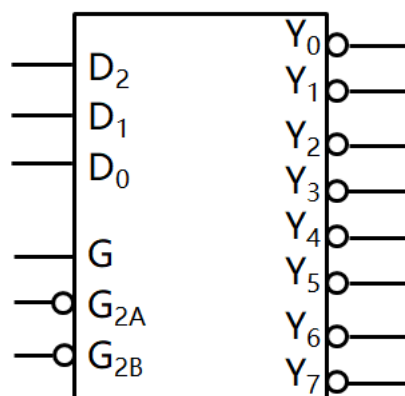


图 1-1 74LS138 译码器

74LS138 具有 6 个输入端和 8 个输出端。输入端 D_0 、 D_1 、 D_2 组成了 8 种编码，每种编码被译成 8 个输出端中的某一个上的低电平信号。输入 G 、 G_{2A} 和 G_{2B}

是三个使能端口，只要有一个使能端口无效，则译码器输出无效；当全部使能端有效时，才对输入进行译码。74LS138 的真值表如表 1-1 所示。

表 1-1 74LS138 译码器的真值表

输入						输出							
G	G _{2A}	G _{2B}	D ₂	D ₁	D ₀	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

2. 2-4 译码器 —— 74LS139

74LS139 是 74 系列集成电路芯片中的一种，实现了 2:4 译码器的功能，如图 1-2 所示。

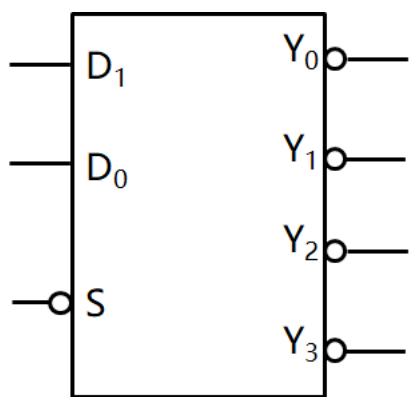


图 1-2 74LS139 译码器

74LS139 具有 3 个输入端和 4 个输出端。输入端 D_0 和 D_1 构成了 4 种编码，每种编码被译成 4 个输出端中的某一个上的低电平信号。输入 S 是使能端口，当其无效时，则译码器输出无效；当其有效时，才对输入进行译码。74LS139 的真值表如表 1-2 所示。

表 1-2 74LS139 译码器的真值表

输入			输出			
S	D_1	D_0	Y_3	Y_2	Y_1	Y_0
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

四 . 实验内容

假如有五个举重裁判，举重选手完成比赛以后，当有多数裁判认定成功时，则成功；否则失败。本次实验请设计此举重裁决电路，即一个 **5 输入的多数表决**

器。该电路的顶层模块如图 1-3 所示，输入/输出端口如表 1-3 所示。使用拨动开关来模拟裁判的裁定，使用 LED 灯来显示是否成功。



图 1-3 5 输入多数表决器电路的顶层模块

表 1-3 输入/输出端口

端口名	方向	宽度（位）	作用
$l_4 \sim l_0$	输入	5	连接拨动开关 SW0~SW4, 用于模拟五个裁判的输入。
led	输出	1	连接 LED 灯 LD0, 用于显示是否成功。灯亮, 表示成功; 灯灭表示失败。

5 输入多数表决器电路的真值表如表 1-4 所示。

表 1-4 5 输入多数表决器电路的真值表

输入					输出
l_4	l_3	l_2	l_1	l_0	pass
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0

0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1

1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

本实验分为两阶段任务，每个阶段均是完成一个 5 输入多数表决器的设计，但采用的设计方法不同。具体实验内容如下所示：

阶段 1：（基于集成电路模块）

1. 根据表 1-1 和 1-2，采用 SystemVerilog HDL 的行为建模方法，完成 74LS138 和 74LS139 两种译码器的设计。
2. 根据图 1-3 和表 1-4，基于 SystemVerilog HDL 的结构化建模方法，调用 74LS138 和 74LS139 两种译码器，以及若干基本逻辑门，完成 5 输入多数表决器电路的设计（类似搭积木），并基于 Vivado 完成行为仿真、综合、实现、生成比特流文件等操作，最终在远程 FPGA 硬件云平台上完成功能验证。

阶段 2：（基于行为建模）

1. 不使用 74LS138 和 74LS139 芯片，直接使用行为建模的方法完成 5 输入多数表决器电路的设计，并基于 Vivado 完成电路的行为仿真、综合、实现、生成比特流文件等操作，最终在远程 FPGA 硬件云平台上完成功能验证。

五． 实验步骤

阶段 1（基于集成电路模块）的实验步骤如下：

- (1). 打开 Vivado 2018.2 集成开发环境，创建一个名字为“**voter_ic**”的新工程（创建工程的具体步骤可参考实验一）。
- (2). 添加第 1 个 SystemVerilog 设计源文件“**dec_74LS138.sv**”。根据表 1-1，采用行为建模方法，完成 3:8 译码器 74LS138 的设计。
- (3). 添加第 2 个 SystemVerilog 设计源文件“**dec_74LS139.sv**”。根据表 1-2，采用行为建模方法，完成 2:4 译码器 74LS139 的设计。
- (4). 根据表 1-4，采用 4 个 74LS138 模块、1 个 74LS139 模块和基本逻辑门电路实现 5 输入多数表决器（这步不是在 Vivado 中完成，而是给出设计方案）。
- (5). 继续在工程 voter_ic 中添加第 3 个 SystemVerilog 设计源文件“**voter5.sv**”。然后，根据第(4)步的设计方案，采用结构化建模方法，通过实例化 74LS138 模块、74LS139 模块以及基本逻辑门完成 5 输入多数表决器的设计。
- (6). 编写测试程序“**voter5_tb.sv**”，对 5 输入多数表决器进行行为仿真，验证其逻辑功能是否正确。
- (7). 如果仿真通过，则继续完成添加约束文件、综合、实现和生成比特流等设计流程（可参照“基于 Vivado 和远程硬件平台的数字逻辑电路设计流程”一文）。
- (8). 登录远程 FPGA 硬件云平台，参照“基于 Vivado 和远程硬件平台的数字逻辑电路设计流程”一文，按照如图 1-4 所示，在实验画纸搭建验证平台，然后，将所生成的 bin 文件下载到远程 FPGA 之上完成功能验证。

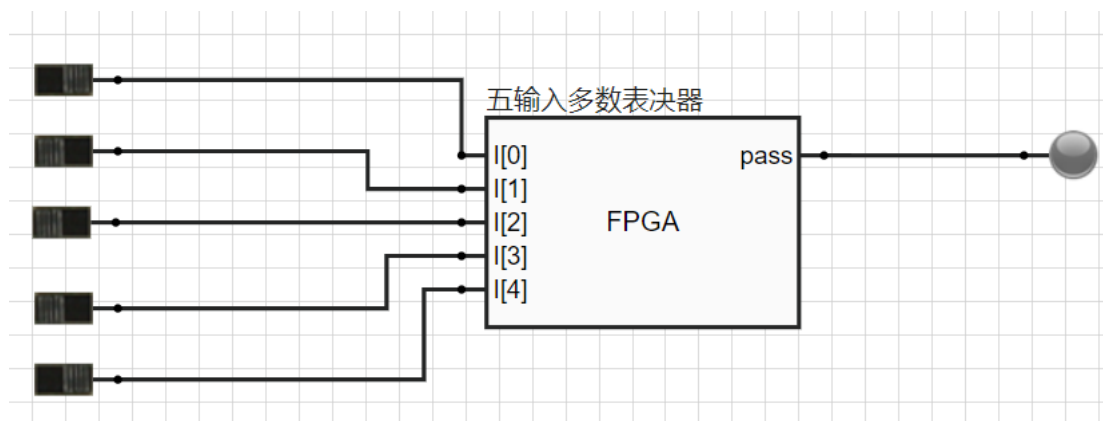


图 1-4 五输入多数表决器验证平台

阶段 2（基于行为建模）的实验步骤如下：

- (1). 打开 Vivado 2018.2 集成开发环境, 创建一个名为“**voter_beh**”的新工程。
 - (2). 添加一个 SystemVerilog 设计源文件“**voter5.sv**”, 采用行为建模的方式完成 5 输入多数表决器电路的建模。
 - (3). 编写测试程序“**voter5_tb.sv**”, 对 5 输入多数表决器进行行为仿真, 验证其逻辑功能是否正确。
 - (4). 如果仿真通过, 则继续完成添加约束文件、综合、实现和生成比特流等设计流程(可参照“基于 Vivado 和远程硬件平台的数字逻辑电路设计流程”一文)。
 - (5). 参照阶段 1, 在远程 FPGA 硬件云平台上完成功能验证。
- 到此, 阶段二的实验已经完成。

六． 实验方式

每位同学独立上机编程实验, 实验指导教师现场指导。

七． 参考内容

1. 教材内容和课件

八．工具下载

课程网盘上下载如下内容：

1. Vivado 2018.2 集成开发环境安装包
2. Vivado 2018.2 安装教程
3. 远程硬件平台管脚对应关系.xlsx
4. 基于 Vivado 和远程硬件平台的数字逻辑电路设计流程.pdf

九．实验报告

1. 写出 74LS138 和 74LS139 的行为建模的 SystemVerilog HDL 代码。
2. 给出基于 74LS138 和 74LS139 的 5 输入多数表决器的设计方案，画出原理图。
3. 写出 5 输入多数表决器的结构化建模的 SystemVerilog HDL 代码。
4. 给出基于行为建模方法的 5 输入多数表决器的 SystemVerilog HDL 代码。
5. 给出远程 FPGA 硬件云平台的验证截图（一组测试向量即可）。
6. 请回答下面这道附加题，并给出答案
 - 只采用 74LS138 译码器和一些基本逻辑门，是否也可以完成 5 输入多数表决器的设计？如果可以，请将设计的原理图贴到实验报告中。

十．篇外话

在本次实验中，我们学习了组合逻辑电路的设计方法，包括行为建模、结构化建模。特别是我们讲解了基于集成电路模块的数字逻辑电路设计。在实际工程设计中，搭建一个复杂的数字系统并不需要完全从零开始进行设计。通常采用的

方法就是选取一些已有的（开源或付费）、已经经过验证的集成电路模块，对其进行集成从而完成最终设计。这种设计方法称为**可重用设计方法学**，而这些集成电路模块称为**IP 核**。

在现代数字集成电路设计中，可重用设计方法学至关重要，也就是说某个模块会被多个数字系统重复使用。支持可重用设计方法学的基础就是 IP 核，即知识产权核（Intellectual Property），它是指某一方提供的、预先设计好的并已经经过验证的、具有某种确定功能的可重用逻辑模块。设计人员常以 IP 核为基础进行数字系统设计，可以大大缩短设计所需的周期，也降低了设计难度。因此，现代数字系统的设计人员更多关注的并非模块的设计，而是模块（IP 核）的集成。IP 核主要分为两大类，**软核**和**硬核**。

- **软核：**与具体工艺无关的、可综合的 HDL 代码，其优点是灵活性高、可移植性强，允许用户自配置；缺点是对模块的预测性较低，在后续设计中存在发生错误的可能性，有一定的设计风险。软核是 IP 核应用最广泛的形式。本实验中所设计的 74LS138 和 74LS139 模块就是一种软核。
- **硬核：**与具体工艺相关的、经过了布局布线和验证的网表文件，设计人员不能对其进行修改。其优点是模块具有稳定的性能、正确性得以最大的保障、设计风险几乎为零；缺点是灵活性差、不能修改、使用范围窄。

Vivado 集成开发环境提倡的是积木式的设计，设计者将功能性模块做成一个一个的 IP 核，然后组装起来成为产品。Vivado 本身也提供了很多 IP 核供设计者使用，例如算术运算（乘法、除法、浮点运算等）、信号处理（FFT、DFT）、图像处理、各类存储器等（[在 Vivado 的流程导航栏中选择 PROJECT Manager → IP Catalog](#) 即可调出 Vivado 的 IP 库，查看其所提供的所有 IP 核）。另外，设计

者也可以使用第三方 IP 核来加快设计。当然设计者也可以开发自己的 IP 核，然后在各个工程中调用或提供给第三方使用。