山东大学 计算机科学与技术 学院

数字逻辑 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000130198 | 姓名： 隋春雨 | | 班级： 20.4 |
| 实验题目：**八位寄存器与八位左移寄存器** | | | |
| 实验学时：4 | | 实验日期： 2021-11-26 | |
| 实验目的：  （1）学习基本时序电路的设计方法；  （2）了解寄存器的工作原理和构成；  （3）熟悉EDA工具软件的使用方法。 | | | |
| 硬件环境：  （1）操作系统为WINDOWS XP的计算机一台；  （2）数字逻辑与计算机组成原理实验系统一台；  （3）基本D触发器8个。 | | | |
| 软件环境：  Quartus II 8.1 Web | | | |
| 实验步骤与内容：  （包括设计的逻辑电路，采用的逻辑门，或者是前期采用基本逻辑电路实现的符合逻辑，写出逻辑表达式，结果预期（采用什么方式展示，如采用那几个发光二极管等），以及最终实现的结果（是否与预期的结果一致，若不一致，是什么问题造成的，经过哪些改进，达到了最终的正确结果））   1. 电路原理图：   本实验要求实现八位的寄存器，电路的原理图如下：     1. 分析：   我们知道，在计算机中，我们是用触发器来实现寄存器的，因为寄存器首先的一个目的就是保证结果的稳定存在，触发器在没有输入的时候，或者R S端都是0 ,这几种情况都可以保证结果不变   1. 电路图：      1. 符号图      1. 实验步骤   （1）原理图输入：根据图3.10电路，采用图形输入法在计算机上完成实验电路的原理图输入。  （2）管脚定义：根据图3.1硬件实验平台资源示意图和附录一 平台资源和FPGA引脚连接表完成原理图中输入、输出管脚的定义。  将寄存器的输出q7－q0分别锁定在LD7－0上。  将寄存器的输入d7－d0分别锁定在K7－0上。  将寄存器的输入脉冲cp锁定在单脉冲上。  （3）原理图编译、适配和下载：在QuartusⅡ环境中选择**EP2C8Q208C8器件，**进行原理图的编译和适配，无误后完成下载。  （4）功能测试：改变K7－0的状态，按动一次单脉冲键，LD7－0的显示将与K7－0相对应，若有错则重新调试。  （5）生成元件符号。   1. 八位左移寄存器电路图   寄存器2   1. 电路的说明：   移位寄存器的形式是串行输入，并行输出。  （1）K0控制要输入的数字：高电平表示输入1，低电平表示输入0，  （2）K1打入一个时钟脉冲，表示输入一个数；  （3）K2为控制移动方向：低电平为左移，高电平为右移，也可以随便输入几个数字进行移动，按K1的结果快一点   1. 分析：我们使用的八位左移寄存器实现的是逻辑左移，就是说，高位被移出去的直接丢弃，低位补0。其中，逻辑移位主要用于无符号数的移位，左移和右移均是移丢一位+空位补0。算术移位主要用于有符号数的移位运算，移位操作时，无论是正数还是负数，无论是原码，反码，补码，均不能改变符号位的位置和值。算术移位分正数的原码，反码，补码的左移和右移，负数的原码，反码，补码的左移和右移 2. 测试结果：左移的例子   输入是1cp1cp0cp1cp1cp1cp | | | |
| 结论分析与体会：   1. 八位寄存器的原理是触发器，在计算机中，我们是用触发器来实现寄存器的，因为寄存器首先的一个目的就是保证结果的稳定存在，触发器在没有输入的时候，或者R S端都是0 ,这几种情况都可以保证结果不变 2. D触发器具有记忆功能、有两个稳定状态，在一定的外界信号作用下，可以从一个稳定状态翻转到另一个稳定状态。触发方式有电平触发和边沿触发两种，前者在时钟脉冲CP=1时触发，后者多在CP的前沿（正跳变0→1）触发。移位都是让前一位的输出是后一位的输入。 3. **分析：我们使用的八位左移寄存器实现的是逻辑左移，就是说，高位被移出去的直接丢弃，低位补0。其中，逻辑移位主要用于无符号数的移位，左移和右移均是移丢一位+空位补0。算术移位主要用于有符号数的移位运算，移位操作时，无论是正数还是负数，无论是原码，反码，补码，均不能改变符号位的位置和值。算术移位分正数的原码，反码，补码的左移和右移，负数的原码，反码，补码的左移和右移** 4. **善用搜索引擎，如果pin栏找不到了，右键找到show选项即可**      1. **在进行电路图绘制之前应该先推导一下表达式，如果没有搞明白原理就开始做实验可能做到一半就不知道做什么了，事倍功半** 2. **通过 Ouartus 日与实验系统的配合，能够在 Pc 环境上实现电路原理图的设计与测试，便于相关知识的学习与练习； 3、对实验中所实现的电路，可进行封装，生成原件符号，提升抽象，利于大型器件的开发与使用；** 3. **我们如果更换了几台箱子发现还是没有工作，可能是PC的问题或者是电源的问题，这次我们发现就是电源有问题导致箱子不工作** 4. **同一个电路图可能有不同的实现方法，我们的原则是选择简洁的，函数成本低的** | | | |