

Lab12: 4 位移位寄存器的设计与实现

实验介绍

这个实验将指导你通过使用 ISE 软件进行 4 位移位寄存器的设计与实现。

实验目标

- 使用 ISE 软件设计并仿真。
- 学会程序下载。

实验步骤

1. 编写文本文件并编译
2. 软件仿真
3. 进行硬件配置

实验原理

1. ISE 软件是一个支持数字系统设计的开发平台
2. 用 ISE 软件进行设计开发时基于相应器件型号的。
注意：软件设计时选择的器件型号是与实际下载板上的器件型号相同。
3. 图 12-1 所示为 4 位移位寄存器的逻辑图，本实验中用 Verilog 语句来描述。

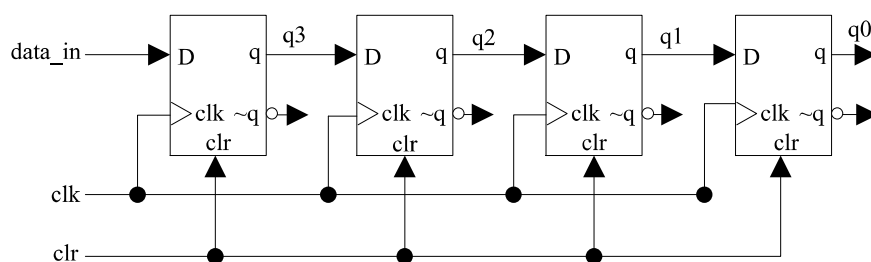


图 12-1

Step 1 新建工程

双击桌面上“Xilinx ISE 12.3”图标，启动 ISE 软件(也可从开始菜单启动)。每次打开 ISE 都会默认恢复到最近使用过的工程界面。当第一次使用时，由于还没有历史工程记录，所以工程管理区显示空白。选择 File → New Project 选项，在弹出的对话框中输入工程名称并指定工程路径。

点击 **Next** 按钮进入下一页，选择所使用的芯片及综合、仿真工具。计算机上安装的所有用于仿真和综合的第三方 EDA 工具都可以在下拉菜单中找到。在图中我们选用了 **Spartan6 XC6SLX16** 芯片，采用 **CSG324** 封装，这是 **NEXYS3** 开发板所用的芯片。另外，我们选择 **Verilog** 作为默认的硬件描述语言。

再点击 **Next** 按钮进入下一页，这里显示了新建工程的信息，确认无误后，点击 **Finish** 就可以建立一个完整的工程了。

Step 2 设计输入和代码仿真

在工程管理区任意位置单击鼠标右键，在弹出的菜单中选择 **New Source** 命令，选择 **Verilog Module** 输入，并输入 **Verilog** 文件名。

单击 **Next** 按钮进入端口定义对话框。其中 **Module Name** 栏用于输入模块名，这里是 **shiftreg**，下面的列表框用于端口的定义。**Port Name** 表示端口名称，**Direction** 表示端口方向(可选择为 **input**、**output** 或 **inout**)，**MSB** 表示信号最高位，**LSB** 表示信号最低位，对于单信号的 **MSB** 和 **LSB** 不用填写。当然，端口定义这一步我们也可以略过，在源程序中再行添加。

定义了模块的端口后，单击 **Next** 进入下一步，点击 **Finish** 完成创建。这样，**ISE** 就会自动创建一个 **Verilog** 模块的模板，并且在源代码编辑区打开。简单的注释、模块和端口定义已经自动生成，接下来的工作就是将代码编写完整。

输入代码后，我们还需要对模块进行测试。在工程管理区将 **view** 设置为 **Simulation**，在任意位置单击鼠标右键，并在弹出的菜单中选择 **New Source**，在类型中选择 **Verilog Test Fixture**，输入测试文件名，单击下一步。这时所有工程中的模块名都会显示出来，我们选择要进行测试的模块。点击 **Next**，再单击 **Finish** 按钮，**ISE** 会在源代码编辑区自动生成测试模块的代码。我们看到，**ISE** 已经自动生成了基本的信号并对被测模块做了例化。我们的工作就是在 **initial...end** 块中的“**//Add stimulus here**”后面添加测试向量。

完成测试文件编辑后，确认工程管理区中 **view** 选项设置为 **Simulation**，这时在过程管理区会显示与仿真有关的进程。右键单击其中的 **Simulate Behavioral Model** 项，选择弹出菜单中的 **Process Properties** 项，会弹出属性设置对话框，其中 **Simulation Run Time** 就是仿真时间的设置，可将其修改为任意时长。

仿真参数设置完后，就可以进行仿真。首先在工程管理区选中测试代码，然后在过程管理区双击 **Simulate Behavioral Model**，**ISE** 将启动 **ISE Simulator**，可以得到仿真结果，如图 12-2 所示。

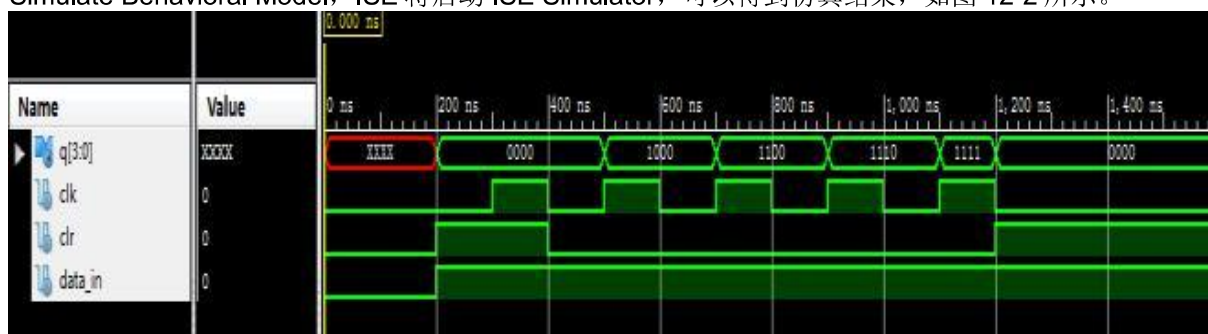


图 12-2

Step 3 综合与实现

在工程管理区的 **view** 中选择 **Implementation**，然后在过程管理区双击 **Synthesize-XST**，就可以开始综合过程。

另外，要实现设计，还需要为模块中的输入输出信号添加管脚约束，这就需要在工程中添加 UCF 文件。在工程管理区单击鼠标右键，点击 **New Source**，选择 **Implementation- Constraints File**，出现一个空白的约束文件，我们就可以为设计添加各种约束。如果综合步骤没有语法错误，XST 能够给出初步的资源消耗情况，点击 **Design Summary**，即可查看。

在过程管理区双击 **Implementation Design** 选项，就可以自动完成实现步骤。如果设计没有经过综合，就会启动 XST 完成综合，在综合后完成实现过程。经过实现后能够得到精确的资源占用情况。在 **Design Summary** 即可看到具体的资源占用情况。

Step 4 器件配置

硬件配置是 FPGA 开发最关键的一步，只有将 HDL 代码下载到 FPGA 芯片中，才能进行调试并最终实现相应的功能。首先我们必须生成能下载到硬件中的二进制比特文件。双击过程管理区的 **Generate Programming File**，ISE 就会为设计生成相应的二进制比特文件。

然后利用 USB-MiniUSB 缆线，来为开发板提供电源和数据下载。我们只需上网下载免费的 Digilent Adept 软件，即可快速实现 Nexys3 开发板上 FPGA 的配置。用 USB-MiniUSB 缆线连接开发板和 PC，打开开发板的电源开关，然后启动 Digilent Adept 软件。系统开始自动连接 FPGA 设备，成功检测到设备后，会显示出 JTAG 链上所用芯片。

界面上将显示检测到 NEXYS3 开发板上的器件 FPGA(XC6SLX16)。这里我们对 FPGA 进行配置。在 **Browse** 中找到之前生成的设计的二进制比特文件，并点击旁边的 **Program** 按钮，软件就开始对 FPGA 进行配置。配置成功后，下面的状态栏会显示 **Programming Successful**。至此，器件配置成功，我们就可以在器件上验证预期的设计有没有很好的得以实现。

Lab5: 7 段显示管的设计与实现

实验介绍

这个实验将指导你通过使用 ISE 软件进行 7 段显示管的设计与实现。

实验目标

- 使用 ISE 软件设计并仿真。
- 学会程序下载。

实验步骤

1. 编写文本文件并编译
2. 软件仿真
3. 进行硬件配置

实验原理

1. ISE 软件是一个支持数字系统设计的开发平台
2. 用 ISE 软件进行设计开发时基于相应器件型号的。

注意：软件设计时选择的器件型号是与实际下载板上的器件型号相同。
3. 在 2 个 7 段显示管上显示一个 2 位的十六进制数，本实验中用 Verilog 语句来描述。

Step 1 新建工程

双击桌面上“Xilinx ISE 12.3”图标，启动 ISE 软件(也可从开始菜单启动)。每次打开 ISE 都会默认恢复到最近使用过的工程界面。当第一次使用时，由于还没有历史工程记录，所以工程管理区显示空白。选择 **File** → **New Project** 选项，在弹出的对话框中输入工程名称并指定工程路径。

点击 **Next** 按钮进入下一页，选择所使用的芯片及综合、仿真工具。计算机上安装的所有用于仿真和综合的第三方 EDA 工具都可以在下拉菜单中找到。在图中我们选用了 **Spartan6 XC6SLX16** 芯片，采用 **CSG324** 封装，这是 **NEXYS3** 开发板所用的芯片。另外，我们选择 **Verilog** 作为默认的硬件描述语言。

再点击 **Next** 按钮进入下一页，这里显示了新建工程的信息，确认无误后，点击 **Finish** 就可以建立一个完整的工程了。

Step 2 设计输入

在工程管理区任意位置单击鼠标右键，在弹出的菜单中选择 **New Source** 命令，选择 **Verilog Module** 输入，并输入 Verilog 文件名。

单击 **Next** 按钮进入端口定义对话框。其中 **Module Name** 栏用于输入模块名，这里是 **x7seg**，下面的列表框用于端口的定义。**Port Name** 表示端口名称，**Direction** 表示端口方向(可选择为 **input**、**output** 或 **inout**)，**MSB** 表示信号最高位，**LSB** 表示信号最低位，对于单信号的 **MSB** 和 **LSB** 不用填写。当然，端口定义这一步我们也可以略过，在源程序中再行添加。

定义了模块的端口后，单击 **Next** 进入下一步，点击 **Finish** 完成创建。这样，ISE 就会自动创建一个 **Verilog** 模块的模板，并且在源代码编辑区打开。简单的注释、模块和端口定义已经自动生成，接下来的工作就是将代码编写完整。

Step 3 综合与实现

在工程管理区的 **view** 中选择 **Implementation**，然后在过程管理区双击 **Synthesize-XST**，就可以开始综合过程。

另外，要实现设计，还需要为模块中的输入输出信号添加管脚约束，这就需要在工程中添加 **UCF** 文件。在工程管理区单击鼠标右键，点击 **New Source**，选择 **Implementation- Constraints File**，出现一个空白的约束文件，我们就可以为设计添加各种约束。如果综合步骤没有语法错误，**XST** 能够给出初步的资源消耗情况，点击 **Design Summary**，即可查看。

在过程管理区双击 **Implementation Design** 选项，就可以自动完成实现步骤。如果设计没有经过综合，就会启动 **XST** 完成综合，在综合后完成实现过程。经过实现后能够得到精确的资源占用情况。在 **Design Summary** 即可看到具体的资源占用情况。

Step 4 器件配置

硬件配置是 **FPGA** 开发最关键的一步，只有将 **HDL** 代码下载到 **FPGA** 芯片中，才能进行调试并最终实现相应的功能。首先我们必须生成能下载到硬件中的二进制比特文件。双击过程管理区的 **Generate Programming File**，ISE 就会为设计生成相应的二进制比特文件。

然后利用 **USB-MiniUSB** 缆线，来为开发板提供电源和数据下载。我们只需上网下载免费的 **Diligent Adept** 软件，即可快速实现 **Nexys3** 开发板上 **FPGA** 的配置。用 **USB-MiniUSB** 缆线连接开发板和 **PC**，打开开发板的电源开关，然后启动 **Diligent Adept** 软件。系统开始自动连接 **FPGA** 设备，成功检测到设备后，会显示出 **JTAG** 链上所用芯片。

界面上将显示检测到 **NEXYS3** 开发板上的器件 **FPGA(XC6SLX16)**。这里我们对 **FPGA** 进行配置。在 **Browse** 中找到之前生成的设计的二进制比特文件，并点击旁边的 **Program** 按钮，软件就开始对 **FPGA** 进行配置。配置成功后，下面的状态栏会显示 **Programming Successful**。至此，器件配置成功，我们就可以在器件上验证预期的设计有没有很好的得以实现。