山东大学 计算机科学与技术 学院

计算机组成与设计 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目：ROM实验 | | | |
| 实验学时：2 | | 实验日期：2025/4/29 | |
| 实验目的：  （1）掌握Vivado中ROM的设置，作为只读存储器ROM的工作特性和配置方法；  （2）用文本编辑器编辑coe文件配置ROM，学习以coe格式文件加载于ROM中；  （3）在初始化存储器编辑窗口编辑coe文件配置ROM；  （4）验证FPGA中ROM的功能。  实验中主要掌握三方面的内容：   1. ROM的参数设置； 2. ROM中数据的写入，即初始化文件的编写；   3、ROM的实际应用，在FPGA上的调试方法。 | | | |
| 实验软件和硬件环境：  软件环境：  Vivado软件、FPGA实验平台  硬件环境：  1.实验室台式机  2.FPGA服务器，PYNQ-Z2开发板 | | | |
| 实验原理和方法：  ROM的工作原理：  ROM（只读存储器）是一种非易失性存储器，数据在制造过程中被写入，或通过编程工具进行初始化，断电后数据不会丢失。  在FPGA中，ROM通常用于存储固定的数据，如查找表、初始化数据等。本次实验使用的单口ROM，在读取数据时，通过地址信号选择要读取的存储单元，数据信号输出对应存储单元的内容。  Vivado中ROM的配置方法：  使用Vivado的IP Catalog中的Block Memory Generator来创建和配置ROM。  配置参数包括存储器类型（单口ROM）、地址总线宽度、数据总线宽度、存储器深度等。  通过导入.coe文件对ROM进行初始化，指定每个地址单元的初始数据。  .coe文件的格式和作用：  .coe文件用于定义存储器的初始化数据，文件中包含数据的进制声明和数据向量。  数据以十六进制表示，每个数据项用逗号分隔，最后一个数据项以分号结尾。  本次实验中，使用.coe文件将特定的十六进制数据加载到ROM中，这些数据将用于后续的验证过程。 | | | |
| 实验步骤：  （1）创建工程：打开本地安装的Vivado 2022.2，新建项目，选择pynq-z2器件。  fig:  （2）添加实验环境：进入FPGA在线实验环境，点击右上角项目材料下载实验源代码和希冀ip核到本地并解压。  fig:   1. 在Vivado项目中，点击Settings→IP→Repository，将上一步解压后的ip\_repo文件夹的位置添加进IP搜索目录。   fig:   1. 点击Sources窗口中的+，选择 Add or create design sources → Next → Add File, 添加实验源代码文件。   fig:   1. 点击Create Block Design创建一个新的顶层设计，随后点击添加IP核按钮，添加cg\_fpga IP.   fig:  （6）点击IP Catalog，并找到Block Memory Generator，开始配置ROM  fig:  （7）按照如下方法定制单口ROM，并命名为rom0:  1726072681382  1726072733931  （8）：勾选Load Init File，并导入你自己编写的.coe文件，初始化ROM  1726072855947   1. ： 2. .coe 文件格式是 Xilinx 公司开发的，用于描述 FPGA 中存储器初始化数据的文件格式。它通常用于将初始化数据加载到 Xilinx FPGA 的内置存储器中，如 Block RAM (BRAM) 或者只读存储器 (ROM)。.coe 文件格式简单且易于解析，通常由以下几个部分组成：  * 一个典型的 .coe 文件包含两部分内容：   1. **Memory Initialization Radix (基数声明)**：      + 指定数据和地址的进制表示。      + 通常有以下几种选择：bin（二进制），hex（十六进制），dec（十进制），oct（八进制）。   + 例如：   + memory\_initialization\_radix=16;   + 这表示文件中的数据是以十六进制形式表示的。   1. **Memory Initialization Vector (数据向量声明)**：      + 列出要加载到存储器中的数据值。      + 数据项之间用逗号分隔，最后一个数据项以分号结尾。   + 例如：   + memory\_initialization\_vector= 3A, 1B, 2C, 0D, FF;   + 这表示要将数据 3A、1B、2C、0D 和 FF 依次加载到存储器的地址 0 到 4 位置。  .coe 文件的完整示例  * 以下是一个完整的 .coe 文件示例： * memory\_initialization\_radix=16; memory\_initialization\_vector= 3A, 1B, 2C, 0D, FF, 10, 20, 30;   + **memory\_initialization\_radix=16**: 表示数据以十六进制表示。   + **memory\_initialization\_vector**: 具体的数据向量，依次为 3A、1B、2C、0D、FF、10、20、30。  1. ：根据要求完成原理图的输入。每一次上升沿，将地址锁入，数码管6-1将显示ROM中输出的数据。LED6-1显示输入的6位地址值。   fig:    （11）右击Sources下顶层设计图标→Create HDL Wrapper，待Wrapper正确生成后，点击左下方Generate Bitstream，开始综合并生成bit文件。注意：综合前wrapper模块应被设置为顶层（加粗表示），若自动设置错误，需右击wrapper图标点击Set as Top手动设置。  fig:  （12）通过 FPGA 云实验平台，可在线分配远程 FPGA 硬件开发板。首先点击 connect 按钮，然后在下拉菜单中选择任意空闲的开发板，并点击Choose File中选择上一步生成的 \*.bit 文件，后点击 send，即可将本地bit文件烧写至希冀远程FPGA.  注意：评测时使用的是固定的数据，请向 coe 文件中加入如下数据：  000FFC,007FFE,01FFFE,03FFFE,07FFFC,0FFFFC,1FFE38,3FF000,3FC001,7F8000,7F0000,FF0000,FE0000,FE0000,FE0000,FC0000,FC0000,FE0000,FE0000,FE0000,FE0000,7F0000,7F8000,3FC000,3FE001,1FF800,0FFFF8,0FFFF8,07FFFC,01FFFC,00FFFC,003FFC,0003E0;    完成实验任务：  认真阅读前面的内容，用图形编辑设计rom。   1. 要求用Vivado IP 库设计单口ROM，地址总线宽度address[]和数据总线宽度q[]分别为6位和24位。      1. 建立相应的工程文件，设置Block Memory Generator数据参数，ROM配置文件的路径（\*.coe）.      1. 完成原理图的输入，完成综合并生成比特流文件。 2. 下载bitstream文件至实验平台，改变ROM的地址a[5..0]，外加读脉冲，通过数码管比较读出的数据是否与初始化数据(\*.coe中的数据)一致。 3. 记录实验数据，写出实验报告。   在FPGA测试平台上面分析其正确性。  首先，输入是地址，然后根据地址输出ROM的内容，这里输入是1，所以输出第一个内容：007FFE。  下面是输入地址是第0个的情况:    还有输入地址是第二个的情况：  平台测评结果如下： | | | |
| 结论分析与体会：  实验结果验证：  通过数码管显示的输出数据与.coe文件中的初始化数据进行对比，验证了ROM的正确性。如果输出数据与预期一致，说明ROM配置成功，数据读取功能正常。  在实验过程中，确保每个步骤的正确性是关键，特别是.coe文件的格式和内容，以及管脚约束的正确性。  通过实验，我掌握了：  Vivado中ROM的配置和使用方法，理解了单口ROM的工作原理。  如何使用.coe文件对ROM进行初始化，以及如何在FPGA中实现和验证ROM的功能。  通过实践加深了对FPGA开发流程的理解，包括工程创建、IP配置、综合、实现和硬件下载等步骤。 | | | |
| 就编译过程中的write bitstream error处理的：  发现是忘记set as top过程，这样编译的是rom文件，不是design文件。 | | | |