计算机设计与实践

附加题设计报告

|  |  |
| --- | --- |
| 班级： | 1236101 |
| 姓名： | 杨世祺 |
| 学号： | 6120310107 |
| 指导教师： | 徐晓华 |

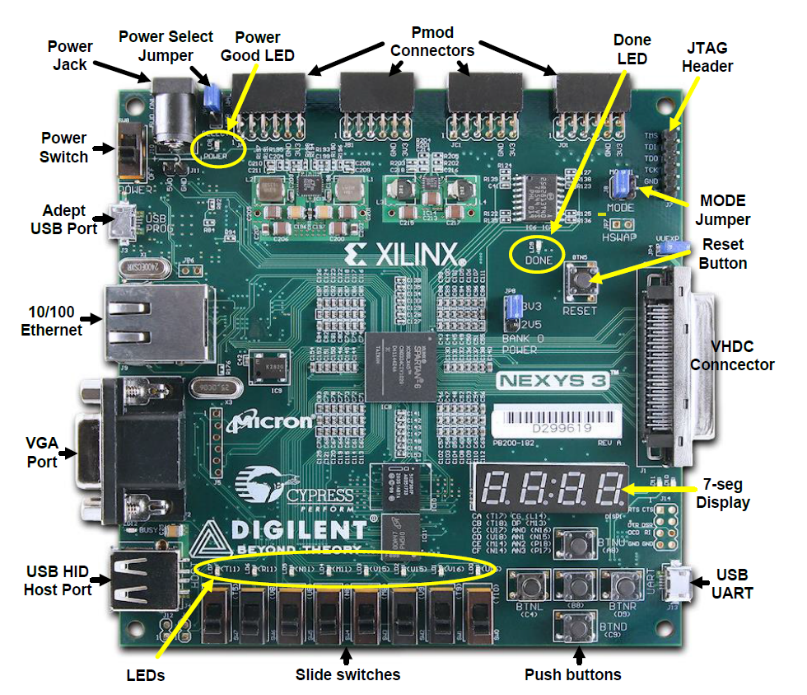
2014年7月19日

计算机设计与实践附加题设计报告

题目：在Nexys 3开发板上实现CPU

1236101班 杨世祺 6120310107

# Nexys 3开发板简述

附加题的要求是在Nexys 3开发板上实现之前设计的RISC处理器。通过阅读开发板的参考手册和电路图可知，Nexys 3开发板上有一片Xilinx Spartan 6 XC6SLX16型号的FPGA，存储器为一片容量为8M × 16的Cellular RAM、一片容量为8M × 16的串行PCM（SPI PCM）和一片8M × 16的并行PCM。其中Cellular RAM结合了SRAM和DRAM的优点，既可以高速访问又可以方便快速地随意寻址，它提供了四种工作方式：异步模式、同步模式、分页模式和突发模式。本实验只使用异步方式，其操作方法与SD2100实验台上的RAM类似。而两片PCM可以当做ROM使用，将指令直接写入SPI中，这样打开开发板电源之后系统能自动将程序下载到FPGA中，同时也可以达到节省RAM空间的目的。

外设方面，Nexys 3开发板提供了一组8位拨码开关、五个按钮、四个七段数码管。此外还带有以下外设接口：一个VGA接口、一个RJ45接口、一个USB 2.0接口，以及一个USB-UART接口，可供开发调试使用。

# 相对于必做实验改动的地方。

## 访存部分

首先给出开发板上存储逻辑的引脚分布，如图2.1所示。结合引脚分布以及三片存储器的特点，可见具体需要做以下改变：

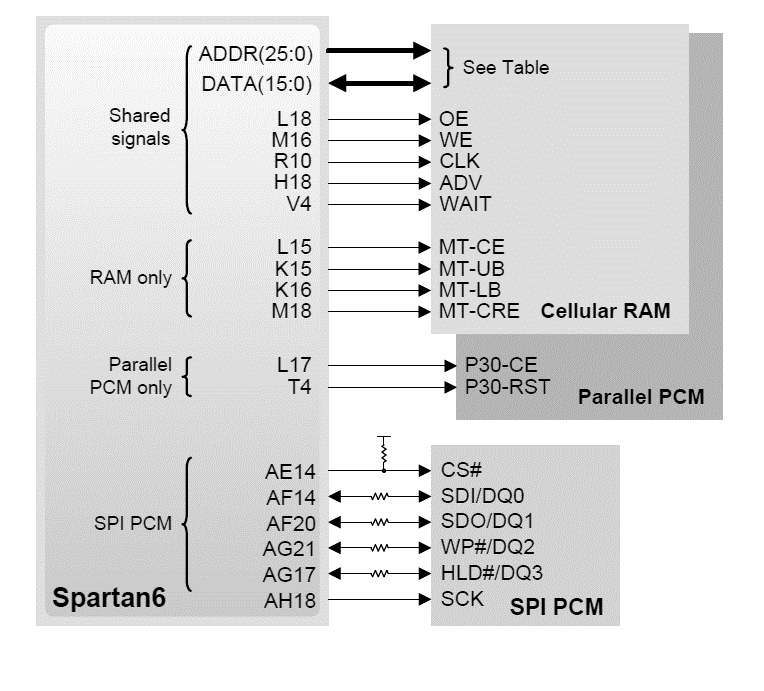


图2.1 Nexys 3 存储逻辑框图

1. 使用Cellular RAM作为数据存储区（用于STA、LDA指令访存等），Parallel PCM（或称为BPI PCM）作为程序存储区（取指令等），SPI PCM用来下载并存储bit文件，以便开机即可将程序下载到FPGA中。
2. 由于存储器容量改变，代码中地址线由16根改为23根（ADDR[25..23]不用于本实验）。
3. Cellular RAM芯片的引脚也有改变，因此访存控制模块需要做相应的修改。在异步模式下使用到的引脚有：MT-CE（RAM片选）、OE（读有效）、WE（写有效）、MT-UB（高位字节有效）、MT-LB（低位字节有效）、CLK（时钟，异步模式下始终置0）、ADV（地址有效，置0）、CRE（控制寄存器有效，本实验不访问，置0），WAIT（等待信号）。
4. 此外，为了从PCM中取指令，还需增加FlashCS和FlashRp引脚用于控制PCM芯片BPI；同时要注意，BPI与RAM共用地址线、数据线和部分控制信号，因此要注意控制好两个芯片的片选信号。
5. 根据Micron公司的存储芯片说明文档，正确使用这片RAM需要初始化、等待、读/写三个状态，其中读写周期如下图所示：

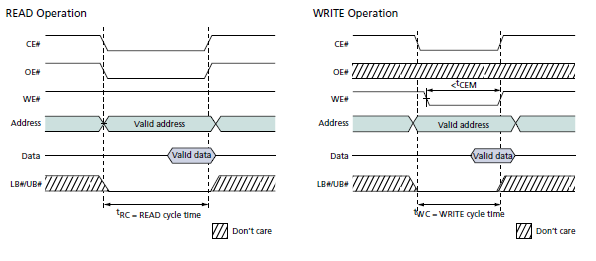


图2.2 Cellular RAM读写时序图

## 显示部分

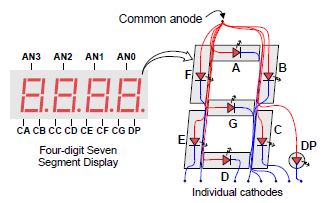
通过Nexys开发板的LED数码管的示意图以及电路图可以看出，出厂时并未给出七段数码管译码逻辑，每一个三极管的点亮需要手动控制；此外四个数码管工作时同一时刻只有一个能被点亮，还需要编写扫描控制逻辑来控制四个数码管的点亮。

图2.3 七段数码管示意图

# 具体实现

## 指令系统部分

该CPU的指令系统与必做实验中的基本相同，除了地址线增加为23根导致PC和访存地址的长度变为23。故不再赘述。

## 访存部分

相比SD2100实验台上的访存部分，这里需要在各个状态将相应的控制字赋值为合适的二进制位，并注意两片存储器的选取。具体为：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **状态** | **CE** | **OE** | **WE** | **LB** | **UB** | **ADV** | **CLK** | **CRE** | **FlashCS** | **FlashRp** |
| 复位 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 空闲 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 取指 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 读 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 写 | 0 | - | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

## 显示部分

首先8位LED灯显示PC的低8位，数码管显示当前的IR值，小数点指示当前节拍。

由于4个数码管共阴极（CA, CB, …, CF, DP），由阳极AN[3:0]负责分时给数码管供电，因此需要实现一个扫描译码逻辑。具体实现为：将开发版上的100 MHz晶振时钟通过分频器降低为100 Hz作为扫描频率（即AN[3:0]变化的频率），然后将16位数据分为4组，在每个时钟的上升沿分别显示4位二进制数。这样根据人眼的视觉暂留效应，看起来就像同时显示16位二进制数一样，同时根据节拍的不同分别点亮4个小数点指示当前节拍。该模块单独实现为一个VHDL文件，具体代码详见hex\_to\_7seg.vhd。

# 下载验证

## 写指令字操作

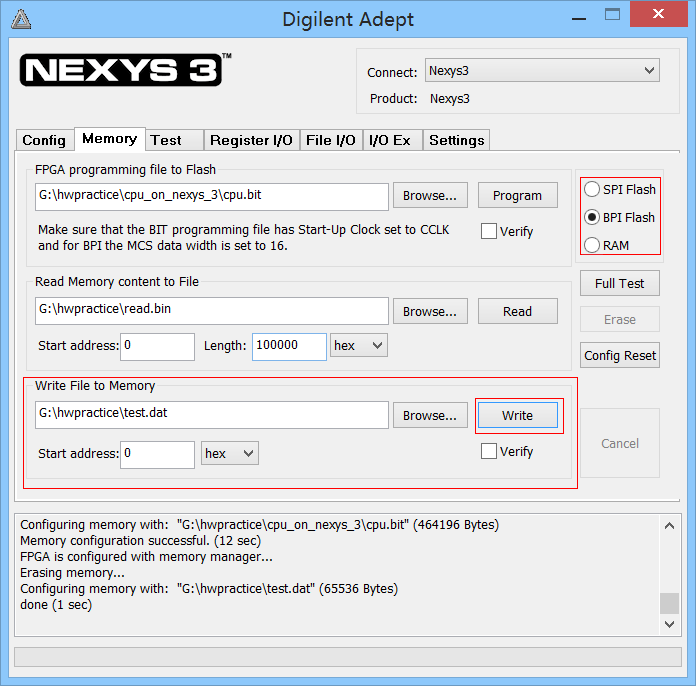
指令是写在BPI PCM中的。因此在Adept程序内执行如下操作：首先将指令按二进制形式写入到一个文件中，然后按照图3.1方式将指令下载到RAM中。

图4.1 将指令写入BPI Flash中

## 下载bit文件

这里分两种情况：

1. 在线下载到FPGA上

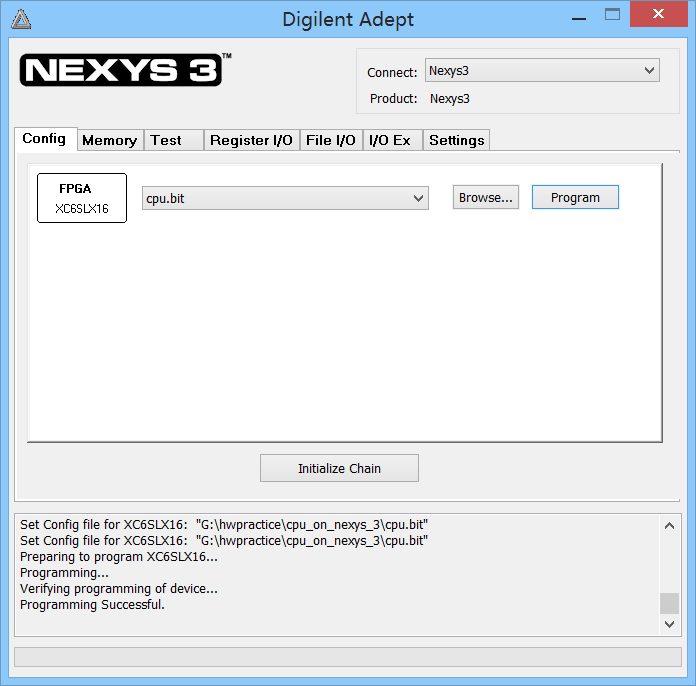
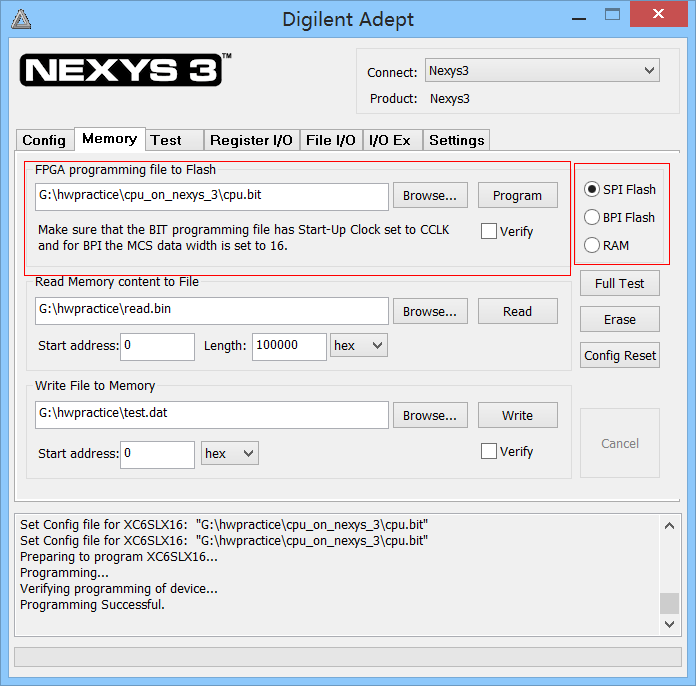
首先在Xilinx ISE中将FPGA Start-up Clock设置为JTAG Clock。之后在Config选项卡中进行下载，如图3.2所示。

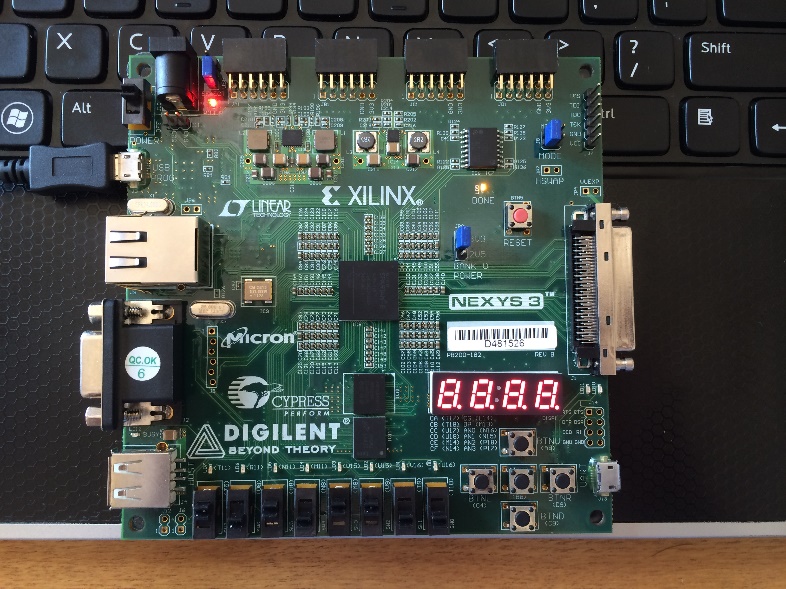
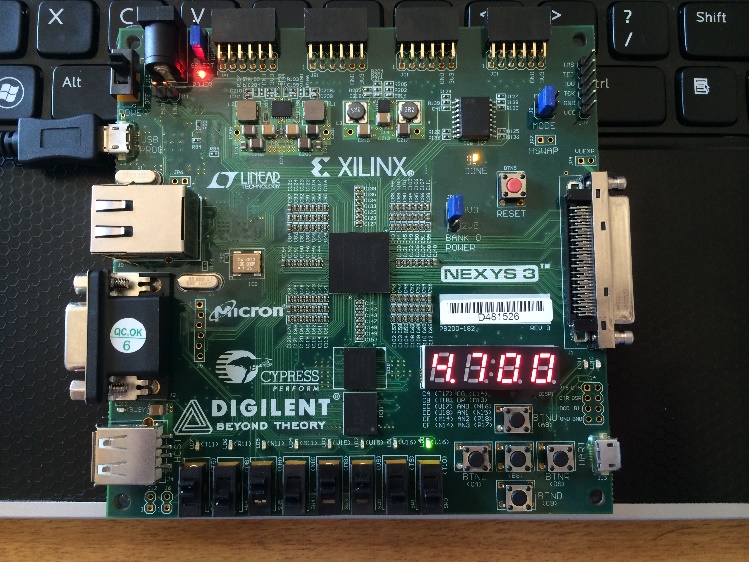
图4.2 将bit文件写入FPGA中 图4.3 将bit文件写入SPI Flash中

1. 下载到SPI PCM中，每次启动自动刷写到FPGA中

首先在Xilinx ISE中将FPGA Start-up Clock设置为CCLK。按照图3.3的设置，在Memory选项卡中，存储器选择SPI，选择bit文件，点击Program进行下载。之后需要按板上的RESET键或者重新开机复位，方可执行程序。

# 实际验证

开机后，系统自动将SPI中的程序下载到FPGA中，此时数码管显示8.8.8.8。按下时钟按钮，则从地址000000h开始读取指令4700，数码管显示4.700，小数点位置为此时所处的节拍，LED指示灯显示00000001。其余指令序列与实验台相同。

图5.1 复位状态 图5.2 第一条指令的取指周期

# 主要问题及解决方案

开始调试的时候，数码管显示的是DBUS的实时值，但是下载后发现无法读取存储器的内容，数据总线一直显示0006（而主存中并未写入任何包含0006的内容）。后来改为显示IR之后所有功能均正常。分析认为是开发板的信号扇出能力不足，不能同时驱动两个引脚信号导致的。解决方法：设置两个输出端口，分别驱动两个管脚即可。