

# 用 PowerPC860 实现 FPGA 配置

■ 郑州通用电控厂 徐以书

## 摘要

介绍如何用 PowerPC860 (MPC860) 进行 FPGA (Xilinx 的 Virtex-II 系列) 的配置; 给出进行 FPGA 配置所需的详细时序图和原理图。本配置基本原理对其它 FPGA 的配置也适用。

## 关键词

PowerPC860 FPGA Xilinx

## 1 概述

MPC860 是基于 PowerPC 结构的通信控制器。它不仅是集成的微处理器, 而且将很多外设的功能也集成在一起。MPC860 具有存储控制器, 其存储控制器的功能很强, 可以支持各种存储器, 包括各种新型的 DRAM 和 Flash, 并可以实现与存储器的无缝接口; 而且使用嵌入式操作系统 VxWorks 和开发环境 Tornado 开发非常方便。

本设计中用 1 片 Intel E28F128J3A150 Flash 作为 BootFlash 对 MPC860 进行加电配置, 其多余的存储空间完全可以存放下 FPGA 所需的配置文件。在加电复位、系统启动后, 由 MPC860 处理机与 EPLD 配合, 控制 FPGA 配置文件的下载过程, 完成对 FPGA 的配置。使用 MPC860 可以对 FPGA 十分方便地进行配置, 甚至可以通过 MPC860 的网络功能对 FPGA 进行远程配置, 节省了因采用专用配置芯片所耗费的电路板及其成本。本文主要介绍如何用 MPC860 对 Xilinx 公司的 Virtex II 系列的 FPGA 进行配置, 其原理同样适用于别的 FPGA 芯片 (包括 Altera 公司)。

## 2 Xilinx FPGA 的配置方式

本设计中 FPGA 采用 Xilinx 公司 Virtex-II 系列的 XC2V4000, 其配置文件的下载模式有 5 种: 主串模式 (master serial)、从串模式 (slave serial)、主并模式 (master selectMAP)、从并模式 (slave selectMAP)、JTAG 模式。其中, JTAG 模式在开发调试阶段使用。本设计将 JTAG 口直接做在信号处理板上, 便于开发设计阶段的调试。

参考 Xilinx 公司的有关文档, 比较其余 4 种下载模式, 可将其分为串行下载方式和并行下载方式。串行下载方式和并行下载方式都有主、从 2 种

模式。主、从模式的最大区别在于: 主模式的下载同步时钟 (CCLK) 由 FPGA 提供; 从模式的下载同步时钟 (CCLK) 由外部时钟源或者外部控制信号提供。主模式对下载时序的要求比从模式严格得多。因此从处理机易于控制下载过程的角度, 选择使用从串模式或从并模式。本设计采用从串模式进行 FPGA 配置, 以减少占用 MPC860 的资源。

用 MPC860 对 FPGA 进行配置, 实质上就是用 MPC860 和 EPLD 来仿真 JTAG 接口的下载时序, 完成对 FPGA 的下载。JTAG 的有效引脚只有 5 个, 分别是 nConfig (PROG\_B)、nStatus (INIT\_B)、Conf\_Done (DONE)、DClk (CCLK)、Data0 (DIN)。其中 nConfig 用于使 FPGA 进入到下载状态, nStatus 用于表示 FPGA 是否已经进入到下载状态以及下载时是否出现 CRC 校验错误, Conf\_Done 用于表示下载完成, 而 DClk 和 Data 则是用于在下载时产生时钟和数据位的。每一个 DClk 时钟周期写入 1 个 bit 数据 (括号中标注为 Xilinx 的 FPGA 对该功能的称呼)。

表 1 所列为下载信号定义。

表 1 下载信号定义

引脚	方向 (对 FPGA)	说明
Data0	输入	数据引脚
DClk	输入	同步时钟, 由外部时钟源或控制信号提供
NConfig	输入	异步复位引脚, 用于异步复位配置逻辑
Conf_Done	输出	配置状态
nStatus	输出	下载状态

## 3 从串模式的配置信号和下载时序

### (1) 启动下载时序

在 nConfig 引脚上产生 1 个超过 2  $\mu$ s 的低脉冲, 等待 nStatus 回应 1 个低脉冲以及 Conf\_Done 变低。

这时候表明 FPGA 已经进入到下载状态，等待至少  $5 \mu s$  以后就可以开始下载了。

### (2) 下载时序

从串下载模式的下载时序如图 1 所示。

在 Data0 上逐位地产生要下载的数据，同时在 DC1k 上产生时钟。数据必须在时钟的上升沿之前  $50 ns$  有效，时钟高电平和低电平的时间都不能小于  $80 ns$ 。这一过程一直持续到全部数据下载完成。在下载过程中可以出现相对较长时间的等待状态，而不会中断下载进程。如果发生错误，nStatus 将被拉低，FPGA 退出操作，必须重复 1~3 步骤。

### (3) 结束下载

在数据都下载完成以后，需要继续给 FPGA 提供若干的时钟，时钟要至少等到 Conf\_Done 变高为止。

### (4) 启动器件

Conf\_Done 被释放，被外部拉高。

MPC860 处理机控制从串下载模式的信号连接示意如图 2 所示。

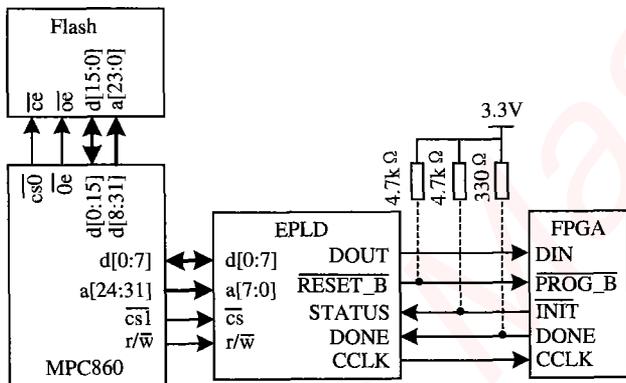


图 2 从串下载模式的信号连接

## 4 MPC860 下载模式的软件（状态机）设计

MPC860 下载模式的状态机的实现是由 MPC860 处理器和 EPLD 协同来完成的。MPC860 通过写入 Sc\_Dnld\_Req(1..0) 来控制状态机的下一个状态，通过判断 EPLD 中寄存器 Sc\_Dnld\_Con(3..0) 的内容来决定流程的跳转。

图 3 中状态 req = “x” 代表向 EPLD 的寄存器 Sc\_Dnld\_Req 写入数据 “xx”。

其中，Sc\_Dnld\_Req (1 downto 0) 使 MPC860 通

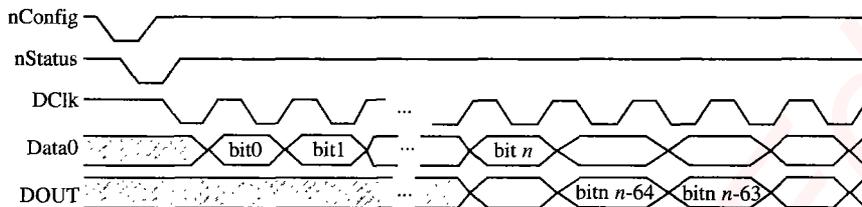


图 1 从串下载模式的下载时序

过这个寄存器向 EPLD 发出指令，控制下载过程：  
10 —— 通知 EPLD 作好向 FPGA 发出 nconfig 的准备工作；

11 —— 通知 EPLD 开始向 FPGA 发出 nconfig 命令；

00 —— 通知 EPLD 作好向 FPGA 下载数据的准备工作；

01 —— 通知 EPLD 开始向 FPGA 逐位下载 1 个字节的

数据。如果下载过程顺利的话，第 1、第 2 步只需在每次下载数据前进行 1 次，第 3、第 4 步是每下载 1 个字节数据都要进行的步骤（每次写入下载数据的操作在第 3 步之前进行，且新数据要写入专门的寄存器 Sd\_Data\_Byte(7 downto 0)）。

Sd\_Data\_Byte(7 downto 0) (下转第 33 页)

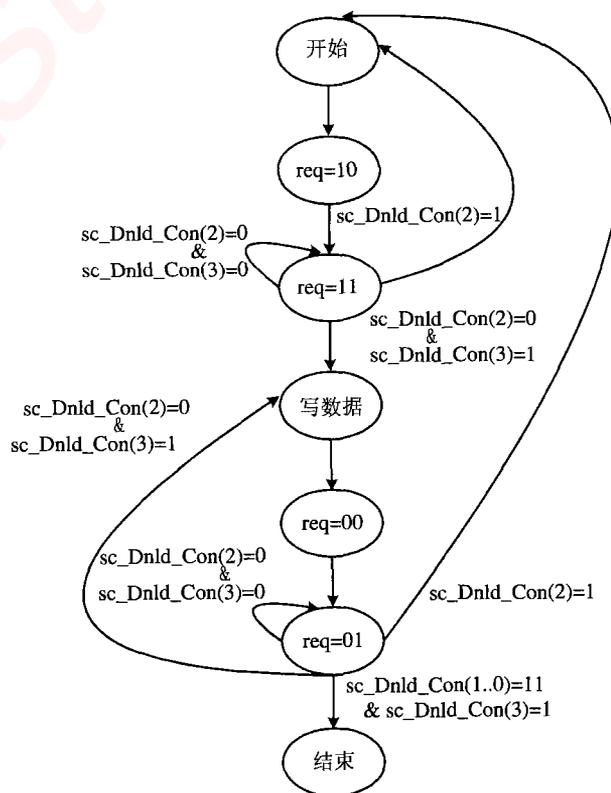


图 3 MPC860 下载模式的状态机

命令格式如下：

```
MOV SFCF, #0C0H ;启动 IAP
MOV SFDT, #55H
MOV SFCM, #01H ;写入命令字
; 查询 SFST.2, 等待芯片擦除结束,
; 用 MOVC 校验是否每个单元均为 FFH。
```

### (2) 块擦除

命令格式如下：

```
MOV SFCF, #0C0H ;启动 IAP
MOV SFAH, #0F0H/#00H ;擦除 block1/block0
MOV SFDT, #55H
MOV SFCM, #00DH ;写入命令字
; 查询 SFST.2, 等待芯片擦除结束,
; 用 MOVC 校验是否每个单元均为 FFH。
```

### (3) 区段擦除

命令格式如下：

```
MOV SFCF, #0C0H ;启动 IAP
MOV SFAH, 区段地址高字节
MOV SFAL, 区段地址低字节
MOV SFCM, #00BH ;写入命令字
; 查询 SFST.2, 等待芯片擦除结束,
; 用 MOVC 校验是否每个单元均为 FFH。
```

### (4) 字节编程

```
MOV SFCF, #0C0H ;启动 IAP
MOV SFAH, 要编程地址高字节
MOV SFAL, 要编程地址低字节
MOV SFDT, 要编程的数据
MOV SFCM, #00EH ;写入命令字
; 查询 SFST.2, 等待芯片擦除结束,
; 校验该单元写入数据是否正确。
```

### (5) 字节校验

```
MOV SFCF, #0C0H
```

(上接第 31 页) 是下载数据缓存。

Sc\_Dnld\_Con (3 downto 0) 是 4bit 的只读寄存器。860 通过查询这个寄存器的内容判断目标 FPGA 的当前下载状态, 以决定应该发出什么样的指令。其各位代表的意义如下所述:

- ① Sc\_Dnld\_Con(3)——为 0 时表示不可以发送新数据到 EPLD, 为 1 时允许发送数据;
- ② Sc\_Dnld\_Con(2)——为 0 时表示工作正常, 为 1 时表示下载过程中断;
- ③ Sc\_Dnld\_Con(1)——目标 FPGA 返回的 nstatus 信号;
- ④ Sc\_Dnld\_Con(0)——目标 FPGA 返回的 conf\_done 信号。

```
MOV SFAH, 要校验数据的地址高字节
MOV SFAH, 要校验数据的地址低字节
MOV SFCM, #06H
NOP
MOV A, SFDT ;A 保存规定单元地址读出的内容
另外, 通过字节编程可以对 Re-Map[1:0]位和 3 个保密位进行编程, 只是在特定的地址写入特定的内容。
```

## 3 SST89C54/58 IAP 功能的用途

通过 IAP 功能可以不需编程器就可做单片机实验。这对初次学习单片机的人们, 尤其对学校单片机的教学带来极大的方便, 是一种最低成本的单片机开发手段。

可以对产品的软件进行升级。不需外加监控芯片, 只是通过串行口便可将 PC 机内的产品升级软件下载到产品中去, 而实现产品软件升级换代。在自己的实验室便可通过 Modem 对远方的产品进行软件升级。这将是以后电子产品的必然趋势。

还可以在线对产品参数进行修改。可实现在线对现场历史数据的存储、曲线参数校正等功能。适用于一些需经常改变数据的应用产品(如利息计、计费器、密码锁及需要升级的产品, 如微型打印机等)及需远距离改变设备参数的产品(遥控报警器、遥控家用电器等遥控设备)。

### 参考文献

- 1 SST 公司. SST89C54/58 单片机用户手册
- 2 薛钧义, 张彦斌. MCS-51/96 系列单片微型计算机及其应用. 第 2 版. 西安: 西安交通大学出版社, 1997

## 结束语

MPC860 有强大的资源, 可以在外围电路不复杂的情况下实现 Xilinx 公司的 FPGA 的配置程序下载。本设计经过实际调试, 已成功地用于所开发的数字通信信号处理板上, 效果良好。

### 参考文献

- 1 Xilinx. Virtex-II Platform FPGA Handbook. 2001
- 2 孔祥营, 柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发系统环境 Tornado. 北京: 中国电力出版社, 2002
- 3 Motorola. MPC860 PowerQUICC User's Manual. 1998

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)

14. [基于 MPC8270 的 VxWorks BSP 的移植](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)

2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)