

基于 FPGA 的在线重构设计方法

王伟伟, 闫新峰, 修展, 徐进, 张甜

(北京航天长征飞行器研究所, 北京 100076)

摘要: 为了解决 SRAM 型 FPGA 和配置存储器中存在的单粒子反转现象导致的 FPGA 工作异常和配置失败问题, 同时满足 FPGA 类产品在总装后可以通过和地面通信接口实现远程更新的需求, 提出了 3 种基于 Xilinx FPGA 的软件重构方法: 对于 7 系列以下 FPGA, 采用 ASIC 在线重构技术, 通过 JTAG 和 SPI 接口控制配置存储器读写实现在线重构, 该方法简单, 适用于多板卡应用场景; 对于 7 系列 FPGA, 采用 MultiBoot 多镜像技术实现多镜像启动, 通过直接更新 M 镜像的方式实现在线重构功能, 该方法不需要外部控制芯片, 电路简单, 设计方便, 适用于单芯片控制场合; 针对 ZYNQ 类芯片, 利用其内部的 BootROM 和 FSBL 功能, 在配置存储器中通过多镜像方式实现在线重构设计, 适用于高性能应用场景; 3 种方法保证了 FPGA 可靠启动, 同时具备在线更新功能, 可以保证在更新失败的情况下, 依然可以运行旧版本或可以重新对其进行配置, 增强了系统的安全性和可靠性。

关键词: FPGA 配置存储器; ASIC; MultiBoot; ZYNQ

Design Method of Online Reconstruction Based on FPGA

Wang Weiwei, Yan Xinfeng, Xiu Zhan, Xu Jin, Zhang Tian

(Beijing Institute of Space Long March Vehicle, Beijing 100076, China)

Abstract: In order to solve the problem of FPGA abnormal work and configuration failure caused by single particle inversion in SRAM FPGA and configuration memory, and to meet the requirement that FPGA products can be updated remotely through the communication interface with the ground after assembly, three software reconstruction methods based on Xilinx FPGA were proposed. For FPGA below 7 series, ASIC online reconstruction technology is adopted, and JTAG and SPI interface control configuration memory read and write to achieve online reconstruction. This method is simple and suitable for multi-board cards application scenarios. For 7 series FPGA, MultiBoot multi-image technology is adopted to realize multi-image startup, and the online reconstruction function is realized by directly updating M image. This method does not need external control chip, and the circuit is simple and the design is convenient, which is suitable for single-chip control situations. Using the BootROM and FSBL functions, ZYNQ chip is designed for online reconfiguration with multiple images in the configured memory, suitable for high performance applications. The three methods guarantee the reliable startup of FPGA and the online update function, which can ensure that the old version can still run or be reconfigured in case of update failure, thus enhancing the security and reliability of the system.

Keywords: FPGA configuration memory; ASIC; MultiBoot; ZYNQ

0 引言

随着航天技术的发展, FPGA 凭借其丰富的接口和逻辑资源, 逐渐取代单片机和 DSP, 越来越多地应用到产品设计中。它可以大大减小设计规模, 提高设计效率。根据可编程技术不同, FPGA 分为 SRAM (static random access memory) 型、反熔丝型和 Flash 型^[1-2]。由于 SRAM 型 FPGA 采用 CMOS 工艺, 集成度高、性能优越、资源丰富、成本低廉, 其可重复编程的特性使它拥有广阔的应用, 尤其是在航天器的在轨更新和维护方面, 优势明显。但航天系统是一个可靠性要求的系统, 空间辐射对于 SRAM 型 FPGA 影响巨大, 会引起单粒子反转, 如果导致关键性能失效, 很有可能导致任务失败, 带来巨大的损失。同时航天工程是一个复杂的工程, 需要各系统协同合作才能完成。通常在研制初期, 由于进度原因, 细节考虑不全或因为用户需求变化, 在总装后需

要对产品中 FPGA 软件进行升级才能正常工作, 这时如果产品不具备在线重构功能, 则必须进行拆卸操作, 由此造成了巨大的时间和成本损失, 因此 FPGA 具备在线重构功能成为系统越来越迫切的需求^[3-4]。

针对 Xilinx 7 系列以下 FPGA、7 系列 FPGA 和 ZYNQ 提出了三种提高在线重构可靠性的设计方法, 分别为基于 ASIC 在线重构设计方法、基于 MultiBoot 多镜像的设计方法和基于 ZYNQ 的在线重构技术^[5-7]。这三种方法几乎涵盖了 Xilinx 所有 FPGA, 实现了在线重构技术, 提高了系统可靠性。

1 FPGA 配置方法

在嵌入式领域, FPGA 凭借丰富的接口和逻辑资源, 越来越受到工程师的欢迎, 尤其是配置方式的多样性和在线重构技术, 给工程师提供了丰富的设计方法和设计空间, 根据实际需求随时改变设计来实现改变或增加新的功能, 提高系统的安全性和可靠性。不同厂家的 FPGA 配置方式有所不同但大同小异, 本文主要针对 Xilinx 的 FPGA 进行研究, 提出设计方法。

Xilinx FPGA 支持多种配置方式, 从配置时钟的来源可

收稿日期: 2020-05-15; 修回日期: 2020-06-03。

作者简介: 王伟伟(1986-), 男, 北京市人, 硕士, 工程师, 主要从事 FPGA 嵌入式设计方向的研究。

以分为：主模式（Master Modes）和从模式（Slave Modes）；从数据读取方式可分为：并行模式（Master Modes）和串行模式（Serial Modes）。根据这两种不同进行组合，可以得到 5 种 FPGA 常用的配置方式：主串、从串、主并、从并和 JATG 模式。工程上最常使用的是 SelectMAP（主并）和 SPI 模式（主从），分开使用 Xilinx 的配置芯片 PROM 和 FPGA 支持的 SPI Flash 芯片。SelectMAP 和 SPI 模式采用主模式，FPGA 在上电后加载片外 Flash 芯片中的配置比特流，配置所需的时钟信号（CCLK）由 FPGA 内部产生，且 FPGA 控制整个配置过程。自动将配置数据从相应的 Flash 芯片读入到 SRAM 中。以 7 系列 FPGA 为例，其配置流程大致分为 8 个步骤，如图 1 所示。

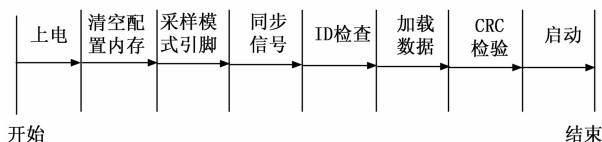


图 1 FPGA 配置流程

上电后，FPGA 的 PROGRAM_B 引脚变低开始整个配置过程，可以通过控制 FPGA 硬件硬件 PROGRAM_B 引脚的变化来重新配置 FPGA。上电后，使用 JTAG JPROGRAM 指令或 IPROG 命令，配置存储器将并清空。在此期间，通过使用全局三态将 I/O 置于高阻态，如果硬件引脚 PUDC_B 为低电平，则内部上拉。硬件引脚 INIT_B 在初始化期间内部驱动为低电平，完成后变为高电平。接着设备对外部模式配置硬件 M[2:0] 进行采样，开始驱动时钟 CCLK，将一组 32bit 特殊的同步字（0xAA995566）发送到配置逻辑。同步字用来将配置数据与内部配置逻辑对齐。设备同步后，先通过设备 ID 检查才能加载配置数据。如果在配置期间发生 ID 错误，则设备会尝试重新配置，设备 ID 检查内置于比特流中。之后 FPGA 开始加载配置帧，加载配置数据时会根据数据包计算循环冗余校验（CRC），然后和比特流中存储好的 CRC 值进行比较，匹配通过后 FPGA 开始进行启动序列，启动完成后，FPGA 配置完成，开始正常工作。

根据上述 FPGA 的配置过程，可以通过软硬件设计保证 FPGA 能够将配置存储器存储的配置比特流可靠地加载到 FPGA 中。随着 FPGA 技术的不断发展，不同代的 FPGA 产品在重构方便拥有不同的特性。Xilinx FPGA 7 系列具备 MultiBoot 多镜像启动功能，支持在线更新系统，配置文件可以现场动态升级。7 系列以前的 FPGA 则不具备该功能，需要外置芯片来进行重构设计。而对于 ZYNQ 系列，由于内置有 ARM 处理器，因此其配置方式更加灵活多样，方便实现在线重构设计^[8-10]。

2 基于 FPGA 的在线重构设计

FPGA 软件的一个重要特性是它具有重构性，可以远程实现配置数据的更新，更改软件，实现新的功能。该功能解决了产品在总装后无法在不拆卸的情况下更新 FPGA

软件的难题，缩短了时间，大大节省了人力财力，提高了效率。针对遥测系统不断变化的需求，软件更新频率很高，因此该功能逐渐变为软件设计的基本要求，增强了产品的安全性和可靠性，大大提高了设计效率。典型的在线升级系统如图 2 所示。

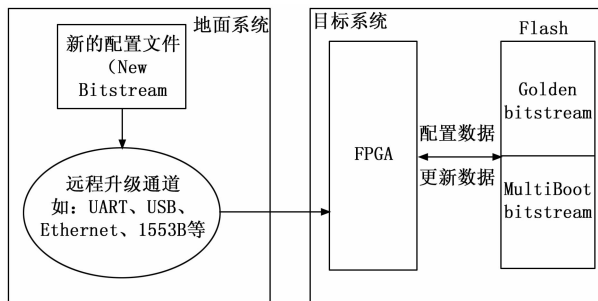


图 2 典型的在线升级系统设计框图

如图 2 所示，一般地，目标系统和地面系统进行通信采用 UART、1553B、Ethernet 等接口形式中的一种或多种，实现弹地数据之间的传输。这样就可以将地面系统中更改后的新的 FPGA 配置文件通过上述接口传输到目标系统上的 FPGA 控制器，接着 FPGA 控制 flash 读写将配置文件正确写入到 Flash 固定位置，从而实现数据流的更新。FPGA 的在线重构技术，关键是要保证在更新失败的情况下，任然可以运行旧版本或可以重新对其进行配置，否则就失去了在线重构的意义。下面针对上文提到的 3 种设计方法分别进行描述。

2.1 基于 ASIC 的 FPGA 在线重构技术

对于 Xilinx 7 系列以下的 FPGA 芯片，本身不具备在线重构能力，需要外置配置管理芯片进行配置，从而实现在线重构，提高可靠性和安全性。一般地，对于只存在一个 FPGA 芯片的设备，可以外加一个 MCU 或 CPLD 来实现对 FPGA 的在线重构设计。Xilinx 的应用笔记中给出了基于 MicroBlaze 软核和 CPLD 的在线重构技术来实现系统的在线更新。若一个设备中有多个 FPGA 芯片，且配置芯片也不同，有采用 SelectMAP 模式的 PROM 芯片，也有基于 SPI 模式的 Flash 存储芯片，各个模块之间通过背板连接。为了减少成本，提高设计效率，可以使用专用 ASIC 芯片来统一管理各 FPGA 芯片，实现在线重构，实现框图如图 3 所示。

该设计可以分别通过 JTAG 和 SPI 接口实现对 PROM 和 SPI Flash 的读写操作，最大支持 8 路 JATG 和 8 路 SPI 接口，同时支持 JTAG 协议的菊花链设计，可以对一个板卡上的多个 FPGA 进行配置。上位机可以通过 UART 接口选择对哪个板卡上的某个 FPGA 进行操作，实现在线升级。同时也可以直接在设备附近使用 JATG 下载器对 FPGA 进行在线调试和升级，这样完全保证了在总装完成后也可以对产品进行调试和升级的需求，解决了软件升级需要重新拆装的难题，大大节约了成本，提高了效率。

2.2 基于 MultiBoot 多镜像的在线重构设计

针对 Xilinx 7 系列 FPGA 可以使用推荐的 MultiBoot 方案来实现 FPGA 的在线重构。MultiBoot 直接操作的是两个

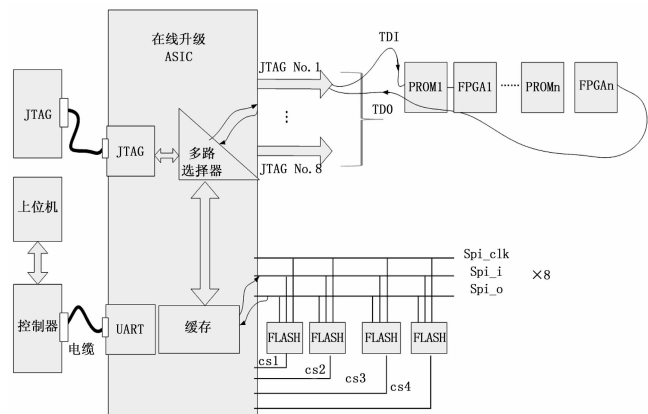


图 3 在线升级 ASIC 设计原理框图

镜像, 也可以是多个镜像。其中一个镜像永远不会改变, 用于令一个镜像在加载失败时可以返回到该镜像上重新加载, 称之为 G 镜像 (Golden Image), 后面的镜像称之为 M 镜像 (MultiBoot Image)。当需要进行远程更新实现重构时, FPGA 可以直接对 Flash 芯片中 M 镜像存放的位置进行读写操作, G 镜像的存储位置和数据永远不变。这样在 M 镜像出现错误时, 可以退回到 G 镜像, 保证 FPGA 能够正常运行, 然后再对 M 镜像进行更新。MultiBoot 实现框图如图 4 所示。

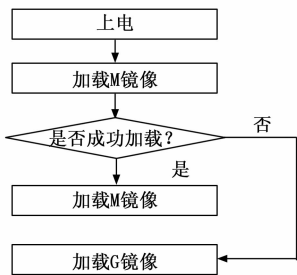


图 4 MultiBoot 实现流程图

一般的 FPGA 从 Flash 的基地址开始存放 G 镜像, 后面存放 M 镜像。根据前面介绍的 FPGA 配置方法中, 上电后, FPGA 从地址为 0 的地方开始加载, 也就是先加载 G 镜像, 但是在 G 镜像的开始位置添加了 IPROG (内部编程) 命令和 M 镜像的地址, 当 FPGA 运行到这个位置时会直接跳到 M 镜像位置开始加载, IPROG 是一条命令可以在 G 镜像中生效, 也可以在设计中使用, 将 IPROG 通过内部配置访问接口 ICAP (internal configuration access port) 接口发送, 从而实现任意时刻触发重新重构。当加载 M 镜像过程中遇到 ID-Code 错误、CRC 错误、WatchDog 超时和 BPI 地址越界错误时, FPAG 会触发退回 (FallBack) 操作, 同时 FPGA 内部寄存器会记录, 发生退回操作后, 会忽略 IPROG 指令, 直接加载 G 镜像后面的配置文件, 运行 G 镜像。

Xilinx 7 系列 FPGA 通过将 IPROG 指令潜入到比特流中的方式, 实现了 FPGA 的多镜像启动, 确保了 FPGA 配置的可靠性。IPROG 指令内容如表 1 所示, 其中 WBSTAR 为 Warm Boot Start Address Register 的缩写, 即热启动地

址寄存器。

表 1 IPROG 指令描述

序号	配置数据	描述
1	0xFFFFFFFF	伪数据
2	0xAA995566	同步字
3	0x20000000	等待
4	0x30020001	往 WBSTAR 中写 1
5	0x00200000	热启动地址: 加载设计的地址
6	0x30008001	往 CMD 寄存中写 1
7	0x0000000F	IPROG 指令
8	0x30008001	等待

在 G 镜像中通过加入 IPROG 指令实现了地址的跳转, 当 M 镜像发生错误后, 会触发 FallBack 操作, 退回到基地址位置, FPGA 将忽略该部分指令, 继续执行 G 镜像的后面其他部分内容, 实现 G 镜像的加载, 正常运行。G 镜像中设计有远程更新程序, 可以操作 Flash 芯片, 更新 M 镜像数据, 从而实现对 M 镜像的在线更新, 实现 FPGA 的远程更新和重构。

2.3 基于 ZYNQ 的在线重构技术

Xilinx ZYNQ 与传统 FPGA 有着巨大的差异, 是一款 SoC (可编程片上系统), 拥有双核 ARM A9 处理器和 7 系列 FPGA 逻辑资源。启动过程体现了以处理器为核心这一特点, 先启动处理器 (PS) 再启动逻辑部分 (PL), 也就是 PL 的配置是通过 ARM 核来配置的, 不支持从 PL 端直接启动配置。

ZYNQ7000 的配置按先后顺序分为 3 个阶段: stage0、stage1、stage2, 其中 stage2 可选。每个阶段的功能及处理过程如表 2 所示。

表 2 ZYNQ 配置过程

阶段	处理过程
stage0	运行 BootROM 中的代码 (不可修改); 1) 读取硬件启动配置引脚 MIO 使能外部对应的非易失存储设备驱动, 包括 SD 卡、NOR Flash、NAND Flash 以及 Quad-SPI 等设备, 并开始搜索启动镜像头文件。根据启动方式不同, 搜索的范围不同; QSPI Flash: 16M 空间; NAND Flash: 128M 空间; Nor Flash: 32M 空间。SD 卡只加载一次, 不会搜索。 2) 读取启动镜像头文件后, 确定启动设备位宽、采用安全配置模式及密钥存储位置等; 3) 根读取信息确定是否采用安全配置模式, 安全配置模式下则启动 PL 侧的安全组件; 4) 读取信息确定代码做 RSA 用户认证; 5) 继续读入存储设备中的数据 FSBL; 6) 对读入数据做相关安全校验, 通过后存入片上存储器 OCM (On-Chip-Memory), 非安全模式下直接导入 OCM; 7) CPU 执行导入 OCM 内的 FSBL, 转入下一阶段 stage1。
stage1	1) 初始化 MIO、PLL (锁相环)、时钟 CLK 等, 在 ps7_init 函数中实现, 完成 PS 端初始化; 2) 判断启动设备, 扫描加载 FPGA 配置文件; 3) 配置 PL 部分, FSBL 通过 PCAP (processor configuration access port) 接口配置 FPGA。 4) 移交给下一阶段, 执行 DDR 中的命令。
stage2	导入操作系统 (可选)。

根据表 2 中的内容可知, ZYNQ 上电后先运行 BootROM, 初始化外部存储设备驱动程序, 然后从存储设备的 0 地址开始, 搜索第一阶段启动程序 FSBL (First Stage Boot Loader), 然后根据 FSBL 加载 PL 镜像和后续的 PS 镜像。基于 Qaud - SPI (QSPI) Flash 启动方式的 ZYNQ 启动流程如图 5 所示。

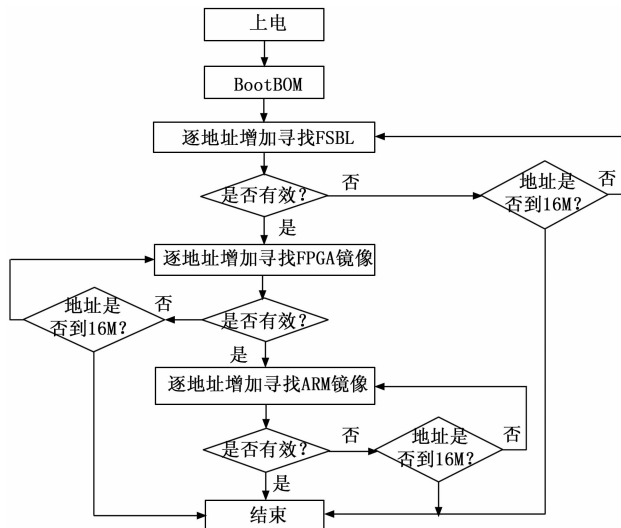


图 5 基于 QSPI Flash 的 ZYNQ 启动流程

如图 5 所示, 上电后, ZYNQ 运行 BootROM, 初始化 QSPI 接口, 开始读取 Flash 中数据, 由于 BootROM 能够访问 QSPI Flash 的最大地址为 16 M, 因此在此地址区间内必须存在完整的 FSBL。找到 FSBL 后, 根据 FSBL 开始加载 PL 和 PS 镜像。由此可知, 如果 QSPI Flash 中存在多个镜像文件时, ZYNQ 只执行第一个镜像, 第一个镜像加载不成功才会加载后面的镜像。根据此特性, 可设计在线升级方案, 如图 6 所示。

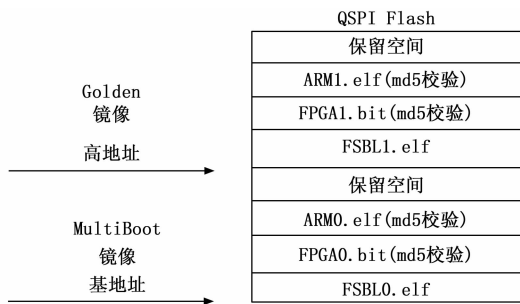


图 6 ZYNQ 多镜像在 QSPI Flash 中的存储位置

如图 6 所示, 在拿到一个全新的 ZYNQ 硬件设计后, QSPI Flash 为空, 不存在任何可执行的镜像文件。这时需要用 JTAG 模式, 将 Golden 镜像下载到 QSPI Flash 的高地址位置, 该地址要大于一个 ZYNQ 可执行镜像的大小, 保证更新时不会破坏到 G 镜像。在生成镜像文件时, PPGA 和 ARM 执行文件要选择 md5 校验, 这样可以确保镜像的正确性和唯一性。在 Golden 镜像中有在线升级程序, 可以根据地面指令, 更新 MultiBoot 镜像, 即从 0 地址开始更新最新的镜像, 由于 G 镜像和 M 镜像中的软件版本有区别, 这样可

以在线验证更新是否成功。如果为成功, 版本信息显示 G 镜像版本; 如果成功则显示 M 镜像版本。通过这种方式可以实现 ZYNQ 类器件的在线重构, 简单方便, 稳定可靠。

3 实验结果和分析

针对 3 种不同的设计方法, 通过设计地面软件来实现对 FPGA 的在线重构。首先打开地面软件, 选择待升级的二进制配置 bin 文件, 并计算总 CRC。然后再给 FPGA 上电, FPGA 在上电后会发送准备好信号给地面软件, 地面收到该信号后会发送在线升级指令给 FPGA, 如果一分钟内未收到在线升级指令, 则 FPGA 退出在线升级状态, 进入正常工作状态。收到在线升级指令后, FPGA 开始对配置寄存器进行擦除操作, 擦除完成后给地面反馈擦除结果。地面收到擦除完成指令后, 开始以每 64 字节为一包发送有效配置数据, 直到所有都完成后, 再发送总 CRC。FPGA 对每包数据进行校验并计算 CRC, 所有包完成后计算得到 CRC, 并和地面传送来的 CRC 进行对比, 正确则烧写成功, 否则烧写失败。烧写成功后, 对 FPGA 重新加电, 通过版本号判断是否更新成功。通过多次测试, 3 种方式均 100% 烧写成功, 可靠实现了在线升级功能。

4 结束语

针对 FPGA 类软件可以在线重构的特性, 根据 Xilinx 三种不同类型的 FPGA 的不同特性, 分别提出基于 ASIC 的 FPGA 在线重构技术、基于 MultiBoot 多镜像的在线重构设计以及基于 ZYNQ 的在线重构技术设计方法。Xilinx 所有的 FPGA, 均可通过这三种技术中的一种或某种组合实现在线重构, 解决了总装后无法进行软件更新的难题, 同时保证了在线重构的安全性和可靠性, 提高了设计效率。

参考文献:

- [1] 刘光文, 李涪阳, 成亮. CPLD 和 Flash 在线升级及配置 FPGA [J]. 微处理机, 2016 (2): 1-3.
- [2] 汪灏, 张学森. 一种基于串行方式的 FPGA 远程加载电路 [J]. 电子科学技术, 2016, 3 (2): 171-173.
- [3] 张永乐, 王永勇, 郑炜. 一种基于 FPGA 的在线程序升级方案 [J]. 嵌入式技术, 2017, 43 (3): 48-54.
- [4] 彭冰, 刘舒, 黄振. 基于共用总线的多片 FPGA 配置电路的设计与实现 [J]. 中国新通信, 2009 (5): 71-75.
- [5] 吴冬冬, 杨晓君, 张佩珩. 一种 FPGA 的远程系统设计方法 [J]. 计算机工程与应用, 2006, 18: 86-69.
- [6] 何宾. Xilinx All Programmable Zynq-7000 Soc 设计指南 [M]. 北京: 清华大学出版社, 2013.
- [7] 何宾, 张艳辉. Xilinx Zynq-7000 嵌入式系统设计与实现 [M]. 北京: 电子工业出版社, 2016.
- [8] 谷鑫, 徐贵力, 王友仁, 等. FPGA 动态可重构理论及其研究进展 [J]. 计算机测量与控制, 2007, 15 (11): 1415-1418.
- [9] 尚丽娜, 徐新民. FPGA 动态重构技术在算术逻辑单元中的应用 [J]. 电子器件, 2007, 30 (3): 365-368.
- [10] 覃祥菊, 朱明程, 张太镒, 等. FPGA 动态可重构技术原理及实现方法分析 [J]. 电子器件, 2004, 21 (2): 277-282.