

# 基于龙芯平台的 EBoot 设计与实现

吴少刚, 丰大强

(中国石油大学(华东)计算机与通信工程学院, 东营 257601)

**摘要:** 在龙芯平台上支持 Windows CE, 对于龙芯处理器在更广泛领域的产业化应用具有重要意义。基于龙芯 2E 开发平台, 设计一个具有可移植性的 Windows CE 的 EBoot 基本架构, 并最终实现 Windows CE 的 EBoot。实验结果表明, 该方案在功能和稳定性上满足 Windows CE 的要求。

**关键词:** 以太网引导程序; 龙芯; WishBone 总线协议; 嵌入式系统

## Design and Implementation of EBoot Based on Godson Platform

WU Shao-gang, FENG Da-qiang

(College of Computer and Communication Engineering, China University of Petroleum(East China), Dongying 257601)

**【Abstract】** Supporting Windows CE on the Godson-based platform, has an important significance to expand the field of industrialization application for Godson processors. This paper presents the design of a portable basic architecture for EBoot of Windows CE, and the implement of the EBoot of Windows CE. Experimental results show that this plan satisfies the functions and stability requirements of Windows CE on the platform for Godson.

**【Key words】** Ethernet Boot(EBoot) loader; Godson; WishBone bus protocol; embedded system

### 1 概述

在特定的硬件平台上开发一款嵌入式系统 Windows CE, Boot Loader 的设计与实现是进行 Windows CE 开发的首要工作。Boot Loader 的主要作用是: 负责把 Windows CE 操作系统映像(OS image)<sup>[1]</sup>文件加载到内存中, 然后执行操作系统映像。在开发和调试 Windows CE 的过程中, Boot Loader 可以方便地将所开发的 Windows CE 系统映像下载到开发板上。与使用 JTAG 直接烧写操作系统映像文件到开发板的模式相比, 基于 Boot Loader 的模式既方便又灵活, 缩短了“修改/构建/下载/调试”的开发周期, 从而大大节省了系统的开发时间<sup>[2]</sup>。

原则上 Boot Loader 的唯一目标是加载并执行 Windows CE 系统的映像文件。Boot Loader 可使用多种方式加载操作系统映像, 例如通过串口、以太网、USB 线等方式来加载。其中最常见 Boot Loader 是通过以太网方式从开发机下载操作系统映像到目标机中, 然后把操作系统映像文件放到内存里执行, 称之为 EBoot(Ethernet Boot Loader)<sup>[1]</sup>。之所以选择 EBoot, 是因为它具有缩短开发时间、操作方便等优点。本文将 EBoot 为例来详细阐述 Boot Loader 的开发, 与使用其他方式进行 Boot Loader 的开发相比, 其基本的流程大致相同。

### 2 龙芯处理器简介

龙芯系列处理器是中国科学院计算所研制的具有自主知识产权的通用处理器, 它具有低功耗、低成本的特点。龙芯 2E 处理器<sup>[3]</sup>是一款实现 64 bit MIPS III 指令集的通用 RISC 处理器。龙芯 2E 的指令流水线每个时钟周期取 4 条指令进行

译码, 并且动态地发射到 5 个全流水的功能部件中。虽然指令在保证依赖关系的前提下进行乱序执行, 但是指令的提交还是按照程序原来的顺序, 以保证精确中断和访存顺序执行。

龙芯 2E 先进的存储系统设计可以有效提高流水线的效率。龙芯 2E 的一级 Cache 由 64 KB 的指令 Cache 和 64 KB 的数据 Cache 组成, 片上二级 Cache 大小为 512 KB, 一级 Cache 和二级 Cache 均采用 4 路组相联的结构。龙芯 2E 通过 24 项访存队列以及 8 项访存失效队列(miss queue)来动态地解决地址依赖, 实现访存操作的乱序执行、非阻塞 Cache、取数指令猜测执行(load speculation)、写合并(store fill buffer)等访存优化技术。

### 3 Windows CE 的开发流程

Windows CE 系统通常针对特定硬件平台定制生成。在定制过程中, 最核心的工作之一就是根据选定的硬件平台和适用的应用场合设计实现板级支持包 BSP(Board Support Package)。

BSP 的开发首先基于硬件平台设计, 要么克隆一个现有的 BSP, 要么开发一个全新的 BSP, 都要设计实现 Boot

**基金项目:** 国家“863”计划基金资助项目“低成本先进计算机”(2006AA010201); 江苏省科技成果转化基金资助项目“龙芯 CPU 产业化应用”(BA2004029)

**作者简介:** 吴少刚(1973 -), 男, 副教授、博士, 主研方向: 计算机系统结构, 嵌入式系统, 并行与分布式计算; 丰大强, 硕士研究生

**收稿日期:** 2009-05-20 **E-mail:** feng\_0917@hotmail.com

Loader; 然后进行 OAL 开发; 之后进行各种设备驱动的开发以及电源管理的开发; 最终得到开发的 BSP, 如图 1 所示。将开发的 BSP 导入到 Windows CE 的开发环境中, 编译成功即可生成 Windows CE 系统的映像文件。

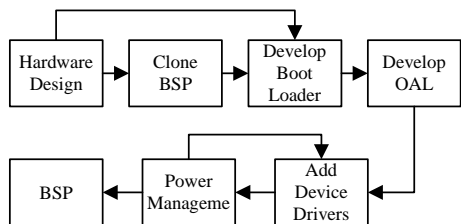


图 1 BSP 的开发流程

## 4 基于龙芯平台的 EBoot 设计与实现

### 4.1 基于龙芯平台的 EBoot 设计

本文的工作主要是基于龙芯开发板平台, 硬件基本配置: CPU: 龙芯 2E 处理器; 北桥: Bonito; 南桥: 威盛 VT82C686B; 网卡: Realtek 8139。

Windows CE 的 EBoot 主要有 3 大功能: 初始化目标硬件设备, 控制启动过程和下载并执行操作系统映像文件<sup>[2]</sup>。根据 EBoot 的主要功能, 将其划分为顺序执行的 2 大阶段: 硬件平台初始化阶段和下载并执行操作系统映像文件阶段。第 1 阶段依赖于 MIPS 体系结构的汇编代码来实现; 第 2 阶段用 C 语言编写, 实现 EBoot 的下载、执行操作系统映像文件的功能。

第 1 阶段是硬件平台初始化阶段。龙芯平台的开发板上电或重置后, CPU 将从 0xBFC00000<sup>[4]</sup>取指令开始执行, 而 ROM 在系统中的地址就是从该地址开始的, 所以, 其中的第 1 条指令就是 CPU 执行的第 1 条指令。其初始化步骤如下:

(1) 确定 CPU 重置的入口地址(由状态寄存器 SR 的 BEV 决定入口地址)。

(2) 跳转到 ROM 中的入口地址开始执行初始化串口、内存、TLB 和 cache、C 语言环境等工作。

(3) 从 ROM 中拷贝代码到 RAM 中, 初始化北桥和南桥, 完成第 1 阶段任务。

第 2 阶段是下载、执行操作系统映像文件阶段。其工作步骤如下:

(1) 全局变量重定位, 通过调用 KernelRelocate() 函数来实现。在很多情况下, EBoot 是在目标设备上的只读存储器上开始运行的, 如果此时 EBoot 要对全局变量进行写操作会导致失败, 则需要把全局变量重定位到 RAM 中来保证全局变量的可写性。

(2) 初始化平台, 通过调用 OEMPlatformInit() 函数来实现。

(3) 以太网下载前的一些准备工作, 如初始化以太网、获得 IP 地址以及初始化 TFTP 服务等, 通过调用 OEMPreDownload() 函数来实现。

(4) 下载操作系统映像文件, 通过调用 DownloadImage() 函数来实现。

(5) 执行操作系统映像文件, 跳转下载的操作系统映像文件的起始地址开始执行, 通过调用 OEMLaunch() 函数来实现, 如图 2 所示。

此时, Windows CE 的 EBoot 的使命已经完成, 系统的控制权将交给下载的操作系统映像文件。

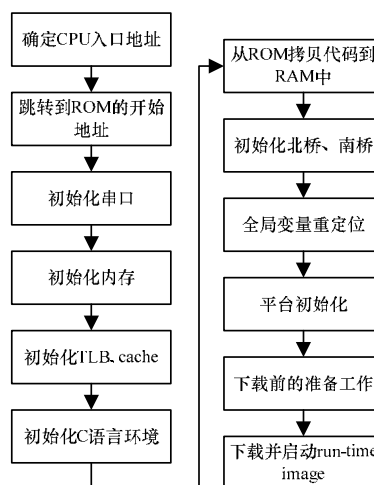


图 2 基于龙芯的 Windows CE 的 EBoot 的架构

### 4.2 基于龙芯平台的 EBoot 的实现

根据龙芯平台的 EBoot 的架构, EBoot 的实现代码流程如图 3 所示。

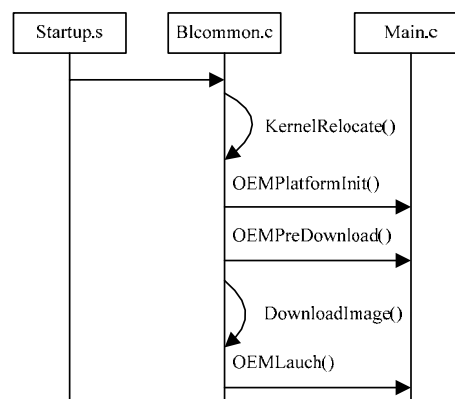


图 3 基于龙芯的 Windows CE 的 EBoot 的代码流程

系统上电或重置后执行的第 1 个函数是 Startup(), 此函数位于 Startup.s 文件中。Startup() 函数负责第 1 阶段的初始化工作, 负责初始化硬件平台, 包括串口、内存、TLB&cache、南桥、北桥等的初始化。Startup() 函数进行基本的平台初始化操作后, 通过跳转语句转移到 Blcommon.c 文件中的主控函数 BootloaderMain() 中执行。

第 2 阶段主要由 Blcommon.c 和 main.c 等文件实现。BootloaderMain() 函数首先执行 KernelRelocate() 函数进行全局变量重定位, 然后执行 OEMPlatformInit() 函数进行平台初始化, 之后执行 OEMPreDownload() 函数进行下载前的准备工作, 然后执行 DownloadImage() 函数来下载操作系统映像文件, 最后执行 OEMLaunch() 函数, 跳转到所下载的操作系统映像文件的起始地址开始执行, 如图 3 所示。

本文是以 EBoot 模式来详细阐述 Boot Loader 的开发, 必须实现对以太网设备的初始化任务。要使基于 PCI 总线的以太网设备能正确的工作, 需要做好 PCI 总线初始化的前置任务。开发者除了需要实现图 3 中 Startup.s 和以 OEM 开头的函数部分之外, 还需要通过建立相关的配置文件才能生成最终的 EBoot。

## 5 关键代码实现

### 5.1 Startup.s 与 blcommon.c

Startup.s 文件是与硬件平台相关的部分, 由开发者自行

实现；main.c 主文件也是由开发者自行实现；而 blcommon.c 文件框架则由微软公司提供，由开发者根据需要进行改造。

## 5.2 main.c 文件

main.c 文件中的 main() 函数实现了 BootloaderMain() 调用的 OEMPlatformInit(), OEMPreDownload(), OEMLaunch() 3 个函数。

(1)OEMPlatformInit() 函数的主要工作是初始化平台。前面的 startup() 函数已经做了基本的硬件平台初始化，该函数主要功能是通过北桥对 PCI 总线进行初始化。

龙芯 2E 北桥<sup>[5]</sup>内部主要模块为：龙芯 2E 的接口(即 SysAD 接口和复位信号)，Local IO 接口模块，中断控制器模块，内部寄存器模块以及 PCI Master 和 PCI target 模块；各个模块之间采用 WishBone 总线<sup>[6]</sup>互连。

CPU 要访问 PCI 上的设备(如网卡)，就需要将 PCI 设备的 memory 空间映射到 CPU 可以访问的 memory 空间，即对 PCI 总线进行初始化。如图 4 所示。

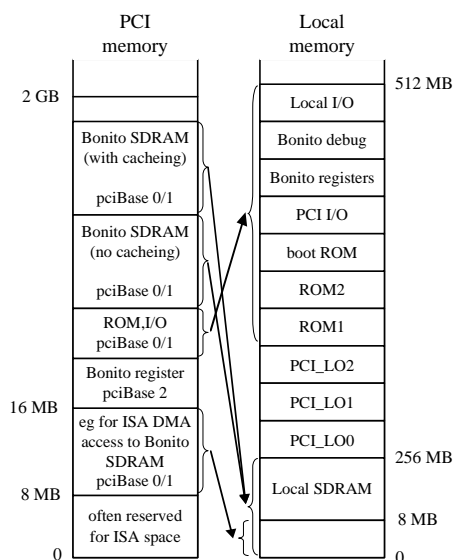


图 4 龙芯平台的本地空间和 PCI 空间的映射

下面是 PCI 总线初始化代码：

```
void OEMPCIInit()
{
    BONITO_PCIMAP =
    BONITO_PCIMAP_WIN(0,
/*PCI_MEM_SPACE_PCI_BASE+*/0x00000000) |
    BONITO_PCIMAP_WIN(1,
PCI_MEM_SPACE_PCI_BASE+0x04000000) |
    BONITO_PCIMAP_WIN(2,
PCI_MEM_SPACE_PCI_BASE+0x08000000) |
    BONITO_PCIMAP_PCIMAP_2; //PCI 的 memory 空间分配；
//memory 空间一共 192 MB，此空间分为 3 部分影射，每部分是
64 MB；
    BONITO_PCIBASE0 = PCI_LOCAL_MEM_PCI_BASE;
//将 PCI 的 memory 空间基地址赋给 BONITO_PCIBASE0；
    BONITO_PCIBASE1 = PCI_LOCAL_MEM_ISA_BASE;
//将 ISA 的 memory 空间基地址赋给 BONITO_PCIBASE1；
    BONITO_PCIBASE2 = PCI_LOCAL_REG_PCI_BASE;
//将 PCI 的寄存器的基地址赋给 BONITO_PCIBASE2；
    /* pci base0/1 can access 256 MB sdram
```

```
/* pci base0/1 可以访问 256 MB 的 SDRAM*/
BONITO_PCIMEMBASECFG = 0;
//PCI 的 memory 空间的基地址的标志位设为 0;
}
```

(2)OEMPreDownload() 函数主要功能是做下载前的准备工作。如初始化网卡、获取 IP 地址、初始化 TFTP 服务等。

(3)OEMLaunch() 函数的主要功能是跳转到下载的操作系统映像文件的起始地址处开始执行。通过跳转语句：JumpTo(launch)，跳转到下载的操作系统映像文件中的第 1 条指令的地址处(通过参数 launch 得到)开始执行；此时，EBoot 的加载并执行操作系统映像文件的任务已完成。

## 5.3 boot.bib 文件

为了能在编译环境中生成 EBoot，还需要通过配置文件 boot.bib 对 EBoot 进行一些配置。下面是基于龙芯平台的 boot.bib 配置文件的部分重要代码：

```
CONFIG
BOOTJUMP=0xBFC0A000
COMPRESSION=OFF
KERNELFIXUPS=ON
```

CONFIG 段主要告知打包工具一些 ROM 信息和配置信息。该段最关键的一项是：BOOTJUMP=0xBFC0A000，它告诉 EBoot 的开始地址在 0xBFC0A000 处。此地址是在 %\_WINCEROOT%\Platform\Godson2e\Target\MIPSII\Retail 目录下生成的 boot.map 文件中 Startup 参数对应的地址得到的。这个地址不是固定的，在查找并替换到此地址后，重新编译即可。

在对 EBoot 进行配置、编译打包后，即可在目录：%\_WINCEROOT%\Platform\Godson2e\Target\MIPSII\Retail 下生成 Windows CE 的 EBoot 二进制文件：boot.bin。

最后，将生成的 EBoot 烧写到开发板中，在目标设备上加电运行 EBoot 就可以加载、调试和执行 Windows CE 系统的映像文件。

## 6 结束语

本文设计的 EBoot 在基于龙芯 2E 平台的 Windows CE 系统移植开发过程中进行了实现，实验结果表明，EBoot 的功能和稳定性满足了 Windows CE 的要求。本文工作对于其他 MIPS 系列处理器开发平台 Windows CE 的 Boot Loader 的开发具有指导意义。下一步工作将继续优化 EBoot 启动性能和 OAL 开发。

## 参考文献

- [1] Platform Builder for Microsoft Windows CE 5.0[Z]. Microsoft Corporation, 2006.
- [2] 何宗键. Windows CE 嵌入式系统[M]. 北京: 北京航空航天大学出版社, 2006.
- [3] 胡伟武, 张福新, 李祖松. 龙芯 2 号处理器设计和性能分析[J]. 计算机研究与发展, 2006, 43(6): 959-966.
- [4] Sweetman D. See Mips Run[M]. 2nd ed. [S. 1.]: Machine Press, 2007.
- [5] BONITO64-"North Bridge" Controller for 64-bit MIPS CPUs[Z]. Algorithmics Ltd., 2001.
- [6] Wishbone System-on-Chip(SoC) Interconnection Architecture for Portable IP Cores Revision B3[Z]. Silicore Corporation, 2001-11.

编辑 索书志