

# 基于 PMON 的龙芯 BIOS 初始化及 VGA BIOS 模拟器

李 雷<sup>1,2</sup>, 郑为民<sup>2</sup>, 刘金刚<sup>1</sup>

(1. 首都师范大学和中国科学院计算技术研究所计算机科学联合研究院, 北京 100037; 2. 中国科学院计算技术研究所, 北京 100080)

**摘 要:** 阐述基于 PMON 扩展后的龙芯基本输入输出系统(BIOS)的初始化流程以及 ATI 显卡 BIOS 的模拟器原理, 针对自检过程中 PCI 设备的初始化、视频图形阵列(VGA) BIOS 模拟器以及北桥的地址空间分配等关键性问题, 给出相应的解决办法。经过调试和测试, 扩展后的龙芯 BIOS 已在基于龙芯 2E 处理器的主板上可靠运行, 能稳定加载为龙芯 2E 移植的 Linux Debian 操作系统内核。

**关键词:** 龙芯; 基本输入输出系统; 视频图形阵列

## Loongson BIOS Initialization and VGA BIOS Emulator Based on PMON

LI Lei<sup>1,2</sup>, ZHENG Wei-min<sup>2</sup>, LIU Jin-gang<sup>1</sup>

(1. Joint School of Computer Science, Capital Normal University & Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100037; 2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

**【Abstract】** This paper elaborates the initial flow of Loongson Basic Input-Output System(BIOS) based on PMON and Video Graphics Array(VGA) BIOS emulator to emulate the initial x86 binaries in ROM of ATI VGA Card. The critical issues such as the initial process of PCI devices, VGA BIOS emulator, and the distribution of North Bridge's address ranges are discussed. After debugging and extensive tests, expanded Loongson 2E BIOS has run reliably in mainboards for Loongson 2E, and can steadily load Debian kernel ported for Loongson 2E.

**【Key words】** Loongson; Basic Input-Output System(BIOS); Video Graphics Array(VGA)

### 1 概述

龙芯二号增强型处理器(简称龙芯 2E)是中国科学院计算技术研究所于 2006 年研制的、具有自主知识产权的 64 位高性能通用 RISC 处理器。它实现了 MIPS III 指令集和一些专用指令, 处理器采用 90 nm 设计技术, 最高主频达到 1.0 GHz, 实测性能超过 1.5 GHz 奔腾 IV 处理器的水平, 具有低成本、低功耗、高性能、高安全性等特点。在不同工作条件下, 龙芯 2E 处理器的功耗在 3 W~8 W 范围内<sup>[1]</sup>。

基本输入输出系统(Basic Input-Output System, BIOS)负责开机时对系统的各项硬件进行初始化设置和测试, 确保系统能够正常工作。若硬件不正常则立即停止工作, 并将出错的设备信息反馈给用户<sup>[2]</sup>。

PMON 是一个兼有 BIOS 和 boot loader 部分功能的开放源代码软件, 多用于嵌入式系统<sup>[3]</sup>。基于龙芯 2E 的系统采用 PMON 作为类 BIOS 及 boot loader, 并做了很多完善工作。例如: 在原来 PMON2000 的基础上添加了硬盘支持, 支持文件系统 ext2 及更多显卡; 修复了 debug 功能, 提高了扩展性。

### 2 龙芯 BIOS 的组成

基于 PMON 扩展的龙芯 BIOS 由以下几部分组成:

(1) 系统自检及设备初始化, 包括 VGA BIOS 模拟器; (2) 系统命令, 包括调试、测试、设置(取消)环境变量、PCI、加载内核等相关命令。

#### 2.1 系统初始化的基本流程

在 32 位 CPU 中, 地址空间段 kseg1: 0xA0000000 ~ 0xBFFFFFFF(512 MB)未被标明也未存入缓存, 即只需将最高 3 位清 0, 这些地址即被映射为物理地址, 无须通过 MMU 转

换, 且不通过 cache 访问。因此, kseg1 是唯一在系统重启时能正常工作的地址空间<sup>[4]</sup>。龙芯 BIOS 就存放在该区间内。

在 64 位的 CPU 中, 地址映射遵循如下规则: 当模拟 1 个 32 位指令集时, 寄存器存放的是对其 32 位进行符号扩展后的 64 位值。因此, 一个 32 位程序访问的是 64 位程序空间内的最低和最高的 2 GB 空间。也就是说, 扩展映射后对最低和最高区域的访问和 32 位情况下一致<sup>[4]</sup>。

当计算机接通电源后, 龙芯 BIOS 将对板上的所有设备进行自检, 包括对 CPU、北桥(中科院计算所自主设计)、内存、缓存、串口、南桥、显卡、软盘、硬盘及键盘进行测试。在系统启动时, 龙芯 2E 的 CPU 从 0x11111111bfc00000 (0xbfc00000 的符号扩展)处开始取指执行, 初始化 CPU 内的寄存器(status 和 cause), 清 TLB, 初始化 TLB; 初始化北桥中的 CPU 和 PCI 总线接口寄存器; 初始化南桥中的 superio, 包括软驱、并口、串口, 串口主要用于错误信息的诊断输出; 初始化内存, 主要通过 I<sup>2</sup>C 协议从内存的 E<sup>2</sup>PROM 读取内存参数来进行设置; 初始化 cache; 测试访问内存是否正确; 将 PMON 的代码拷贝到内存。上述均使用汇编语言, 之后可读写内存并使用栈。

系统在内存中运行时, 入口为 initmips, 并将内存大小作为参数传递给它。执行 initmips 之前, CPU 只作初步初始化, 其作用只是为 CPU 在内存中运行作一些必要的准备。

**作者简介:** 李 雷(1981 - ), 男, 硕士研究生, 主研方向: 计算机体系结构; 郑为民, 副研究员; 刘金刚, 教授、博士生导师

**收稿日期:** 2008-05-12 **E-mail:** lilei06@gmail.com

进入 initmips 之后：

(1)初始化 CPU 流水线频率和总线频率。

(2)初始化标准环境变量表；初始化 PCI 总线和设备；初始化网络；进程号 histno 初始化为 1；符号表初始化，建立“PMON”符号表，它对应的地址是 PMON 程序的入口，建立“start”符号表，它对应的地址是 0x80100000，定义“\_ftext”和“etext”符号表；建立控制台终端初始状态；输出本程序和机器相关的版本和其他信息；如果上次设置自动加载内核，则提取上次设置的变量，在用户按下回车键之后，自动加载位于硬盘分区里的内核。

(3)安装异常向量；清 cache；设置好状态寄存器，即将 Bev 位清零，表示进入正常模式。

(4)显示 PMON 提示符，等待用户输入命令。

初始化流程如图 1 所示。

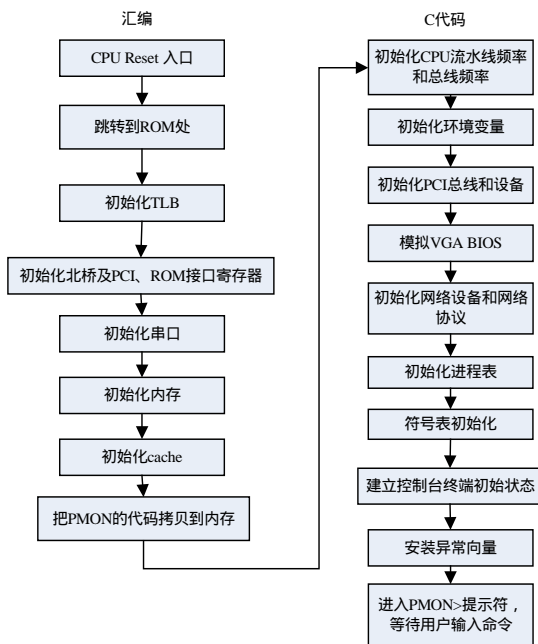


图 1 龙芯 BIOS 初始化流程

## 2.2 系统命令设置

基于 PMON 扩展的龙芯 BIOS 中定义了大量命令。其数据结构主要由命令名称、参数、参数描述、命令描述、命令处理函数、最小参数个数、最大参数个数组成<sup>[5]</sup>。命令主要包括：

(1)h 命令：查看 BIOS 中所有命令及其说明；

(2)load 命令：从硬盘、tftp 服务器上加载内核，还可升级 BIOS；

(3)reboot 命令：重启；

(4)ifaddr 命令：设置 IP 地址；

(5)ping 命令：确定本地主机是否与另一台主机交换数据包；

(6)set 命令：显示和设置环境变量；

(7)unset 命令：取消 set 命令设置的变量；

(8)ifm 命令：设置网络的工作模式，包括 auto, 10 Mb/s, 100 Mb/s；

(9)setmac 命令：设置 MAC 地址。

另外还包括调试命令以及新增加的测试命令等。例如，可以测试 CPU 浮点部件、内存、网络、CPU 的频率、PCI 设备、显示、硬盘、键盘和串口。

## 2.3 引导加载

自检测试完成后，系统将在用户输入正确的加载内核 load 命令<sup>[5]</sup>参数提示下，在给定的驱动器中寻找操作系统，并向内存中装入操作系统。BIOS 可以载入几种格式的文件，包括 elf 可执行文件、二进制 bin 文件。它们分别对应载入函数 load\_bin 和 load\_elf。

## 2.4 VGA BIOS 模拟器

绝大部分显卡通过执行存储在 VGA ROM 里的 x86 二进制代码来进行显卡初始化。在非 x86 平台上，处理器不能直接运行该代码，显卡一般不能被正确初始化而导致不可用，因此，龙芯 BIOS 在 PMON 的基础上扩展了 VGA BIOS 模拟器，模拟显卡 BIOS 运行环境，达到正确初始化显卡的目的。

在 PCI 设备初始化过程中，如果发现设备的类代码为显示控制器，系统先会设置全局变量 vga\_dev，然后调用模拟初始化函数 vga\_bios\_init。它会从设备的扩展 ROM 的 header 处进一步分析是否是 VGA Card。若是，则首先分配结构体变量 xf86Int10InfoPtr，用来存储模拟环境，然后调用执行函数 xf86Int10ExecSetup。

VGA BIOS 模拟包括以下几个部分：显存地址 0xa0000 模拟，显卡 BIOS 地址 0xc0000 模拟，x86CPU 寄存器及指令模拟，中断向量主要是 INT10 模拟和 sys BIOS 模拟等。

现在龙芯 BIOS 已经可以成功初始化的显卡包括：ATI 的 RAGE PRO, RAGE XL, RADEON7000 等。在显卡初始化的过程中，借用了 xfree86 源码和 Linux 内核源码中对 ATI 显卡初始化的经验。

## 3 关键技术

在初始化过程中，比较重要的是函数 tgt\_devinit()，其主要任务就是正确初始化 PCI 总线和 PCI 设备<sup>[6-7]</sup>。

(1)初始化挂接在南桥上的设备<sup>[7-8]</sup>，包括串口、键盘、硬盘、USB。

(2)开启 cache，进行 cache 初始化配置。

(3)调用 \_pci\_businit 初始化 PCI 总线：先调用 \_pci\_hwinit 初始化北桥<sup>[7]</sup>和 PCI 总线 bus0(与北桥直接相连的总线)数据结构 and 地址寄存器，为 PCI 总线定义描述符，并为总线的各个域赋值，其中，主要是为本总线分配起始 memory 地址空间和起始 IO 地址空间；为北桥的 PCIMAP 寄存器(由 PCI\_Lo0, PCI\_Lo1, PCI\_Lo2 组成)赋值，同时对 base0 和 base1 及其各自的 mask 寄存器赋值。

(4)调用 \_pci\_scan\_dev 和 \_setup\_pcibuses 初始化 PCI 设备：先遍历 PCI 总线上的所有 PCI 设备，并为它们分配数据结构，将该设备与其双亲以及兄弟链接起来形成一个树状结构，北桥是这棵树的根。根据设备配置寄存器的值为设备的各个域赋值，并计算本设备所需要的 IO 空间和 memory 空间，将所得的空间大小通知给双亲设备用来做以后的计算。其中包含 2 种遍历方式：从上到下查询设备并建立数据结构；从下到上搜集每个设备所需空间，然后从上到下为每个设备分配空间。

北桥地址空间的具体分配如表 1 所示。须注意 因为 VGA 显存在 PCI 空间的 0xb8000，即低 1 MB 内，所以为访问显存，将 PCI\_Lo0 映射到 PCI memory 空间起始的 64 MB 空间内。PCI\_Lo1 和 PCI\_Lo2 的映射，则是为了操作系统访问的方便，使 PCI 空间地址和 CPU 空间地址相同。另外，为让兼容 ISA DMA 的 PCI 设备通过 DMA 访问系统内存，将内存的 0 MB~16 MB 空间映射到 PCI 空间的 0 MB~16 MB<sup>[6]</sup>。为使其其他 PCI 设备

如网卡等通过DMA访问系统内存，将CPU空间的 2 GB开始空间映射到PCI空间的 2 GB开始空间。

表 1 北桥的地址空间分配

物理地址	空间/Bytes	类别
0000 0000	256 M	MEM_1
1000 0000	64 M	PCI_Lo0
1400 0000	64 M	PCI_Lo1
1800 0000	64 M	PCI_Lo2
1c00 0000	56 M	ROM
1f80 0000	4 M	ROM
1fc0 0000	1 M	Boot
1fd0 0000	1 M	PCI I/O
1fe0 0000	256	北桥本身 PCI 配置寄存器
1fe0 0100	256	北桥内部寄存器
1fe0 0200	256	未使用
1fe8 0000	512 K	PCI 配置空间读写
1ff0 0000	256 K	Local I/O
1ff4 0000	256 K	Local I/O
1ff8 0000	256 K	Local I/O
1ffc 0000	256 K	Local I/O
2000 0000	1.5 G	PCI 地址
8000 0000	2 G	MEM_2

在北桥中关于地址映射的几个重要的寄存器为 Pccimembasecfg, Pccimap, Pccimap\_cfg。其中, Pccimembasecfg 寄存器的格式如图 2 所示。

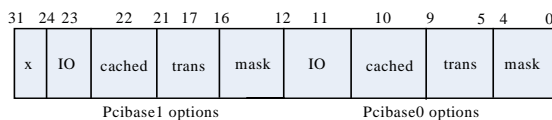


图 2 Pccimembasecfg 寄存器格式

为使 PCI 设备能通过 DMA 访问系统内存，须把 PCI 地址映射成 CPU 地址。通过设置 Base Address Register 完成该过程。北桥共有 3 个这样的寄存器：Pccibase0, Pccibase1, Pccibase2, Pccibase0, Pccibase1 均可以映射多达 256 MB 的空间，Pccimask0, Pccimask1 决定映射的地址空间大小。Pccibase2 映射北桥的内部寄存器，映射区间为 64 KB。

为了让 CPU 访问 PCI 空间，须将 CPU 空间映射到 PCI 空间。在内存空间 256 MB 上方有 3 个连续 64 MB 的空间，即 PCI\_Lo0, PCI\_Lo1, PCI\_Lo2。这 3 个区间可被北桥映射到 PCI 空间以 64 MB 对齐的任意位置。通过设置 PCIMAP 寄存器可以完成。该寄存器的格式如图 3 所示。

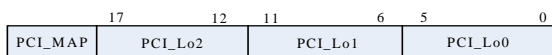


图 3 PCIMAP 寄存器格式

(上接第 203 页)

## 5 结束语

本文设计的基于 M 语言和 LS SIMD 计算机汇编语言的编译系统具有较高目标代码执行效率，对其他媒体处理器的高级语言和编译系统设计具有一定的参考价值。实例验证了编译的正确性以及目标代码的质量，所有的 M 语言程序执行结果和汇编程序执行结果一致。

### 参考文献

[1] 李 莉, 沈绪榜. LS SIMD 阵列微处理器的控制逻辑设计[J].

CPU 访问 PCI 空间时, CPU 的低 26 位地址直接传到 PCI 的低 26 位地址, PCI 的高 6 位地址分别由 PCI\_Lo0, PCI\_Lo1, PCI\_Lo2 代替。Pccimap2 是说明映射到 2 GB 以上的空间还是 2 GB 以下的空间。

CPU 通过访问物理地址 0x1fe80000 即可访问 PCI 设备的配置空间。分为 2 种配置周期: Type0 和 Type1。为了读写 PCI 设备的配置头, 须事先设置 Pccimap\_cfg 寄存器的值, 其格式如图 4 所示。



图 4 Pccimap\_cfg 寄存器格式

AD16UP 是高 16 位地址。当 CPU 访问 PCI 设备的配置空间时, PCI 总线上的地址组成如图 5 所示。

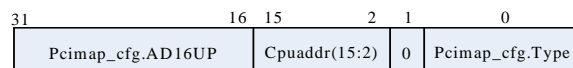


图 5 PCI 总线上的地址组成

## 4 结束语

由于龙芯 BIOS 在 PMON 的基础上改动比较大, 因此调试周期比较长。现在它已能稳定运行于中科院计算所龙芯 2E 移植的 Linux Debian 操作系统中。经过大量测试和实际用户体验表明, 龙芯 BIOS 运行可靠, 可推广应用。

### 参考文献

[1] 中国科学院计算技术研究所. 龙芯 2E 处理器数据手册[Z]. 2006.  
 [2] 韩山秀, 樊晓桢, 张盛兵, 等. BIOS 的设计与实现[J]. 微电子学与计算机, 2005, 22(11): 113-115.  
 [3] Opsycon AB. PMON2000 Manual[Z]. 2000.  
 [4] Sweetman D. MIP 处理器设计透视[M]. 赵俊良, 张福新, 陶 品, 译. 北京: 北京航空航天大学出版社, 2005.  
 [5] Algorithmics Ltd.. PMON Users Manual[Z]. 1998.  
 [6] PCI Special Interest Group. PCI Local Bus Specification(Revision 2.3)[Z]. 2002.  
 [7] PCI-to-PCI Bridge Architecture Specification Revision 1.1[Z]. (1998-12-18). <http://www.pcisig.com>.  
 [8] VIA Technologies, Inc.. VT82C686B South Bridge Data Sheet Revision 2.38[Z]. 2005.

计算机学报, 2000, 23(5): 558-559.

[2] 沈绪榜. MPP 嵌入式计算机设计[M]. 北京: 清华大学出版社, 1999.  
 [3] 李 莉, 钱 刚, 沈绪榜. LS SIMD 协处理器的通讯机制及其 VLSI 实现[J]. 微电子学与计算机, 2002, 19(9): 52-53.  
 [4] Levine J R, Mason T, Brow D. Lex 与 Yacc[M]. 杨作梅, 张旭东, 译. 北京: 机械工业出版社, 2003.  
 [5] 谢丹夏, 程 旭. 面向嵌入式的优化编译器生成器设计[J]. 计算机应用研究, 2001, 18(4): 9-11.