• 嵌入式系统工程 •

龙芯 Mipsel 平台 PMON 开发

钱振江[1]， 常晋义[1]， 张福新[2]

(1. 常熟理工学院 计算机科学与工程学院，江苏 常熟 215500;
2. 中国科学院计算技术研究所 系统结构重点实验室，北京 100080)

摘 要: PMON是一种针对嵌入式系统而设计的类BIOS引导程序。为了在龙芯平台上为各种应用系统提供PMON的支持，从PMON2000的基本架构出发，研究并实现在龙芯Mipsel平台上的PMON2000基本框架。首先对PMON的整体框架进行了认知与分解，指出根据龙芯Mipsel平台的独特性对PMON文件系统目录中的各种系统文件进行移植是设计过程中的难点。在此基础上，利用汇编语言实现对硬件的基本初始化，使用C语言实现对南北桥芯片以及显卡等信息的初始化工作。通过在龙芯系列平台的运行，验证了该设计的可行性和正确性。

关键词: 龙芯处理器； 系统架构； 引导程序； PMON系统； 移植

中图法分类号: TP303 **文献标识码**: A **文章编号**: 1000-7024 (2010) 07-1473-04

Developmentment of PMON based on Loongson Mipsel platform

QIAN Zhen-jiang[1], CHANG Jin-yi[1], ZHANG Fu-xin[2]

(1. School of Computer Science and Engineering, Changshu Institute of Technology, Changshu 215500, China; 2. Key Lab of Computer System and Architecture, Institute of Computing Technology, Chinese Acadmy of Sciences, Beijing 100080, China)

Abstract: PMON is a bootstrap program similar BIOS for embedded systems designed. To provide PMON support for the various application systems on the Loongson platform, the basic framework of PMON for the Loongson Mipsel platform with PMON2000 basic framework is researched and realized. Firstly, the overall framework of PMON is recognized and decomposed. It is pointed out that the difficulty in scheme design is to transplant the variety of system files within the PMON file system directory based on the uniqueness of Loongson Mipsel platform. On this basis, the assembly language is used to achieve the basic initialization of hardware, and the C language is applied for the realization of initializing the south and north bridge chip, as well as graphics. Finally, by running on Loongson series platform, the feasibility and correctness of the presented method is demonstrated.

Key words: Loongson processor; system architecture; bootstrap program; PMON system; transplant

0 引 言

龙芯2号处理器[1-2]是中科院计算所研发的高性能通用处理器，是一种类MIPS[3]指令系统的高性能64位通用CPU芯片。随着龙芯系列CPU的性能不断提高，龙芯已经涉及个人桌面计算机以及高端的服务器应用领域。在计算机的体系结构中，基本输入输出系统BIOS是必不可少的。PMON[4]是一款开源的ROM调试监视器，早期是为LSI Logic MIPS R3000评估板的设计需要而开发的。经过多年的发展，PMON已成为MIPS领域评估板和开发系统的通用固件。目前，最新的PMON2000是PMON的非官方后继版本。龙芯平台采用PMON2000来作为基本输入输出系统。

1 PMON概述

PMON是嵌入式系统[5]的一种引导方式，相当于PC中的BIOS。其功能十分强大，能够从文件系统直接引导加载内核，而且具有一般BIOS不可比拟的优势。

PMON早期版本的功能主要有shell、net、load、debug等，但不支持硬盘，显卡。对于一些嵌入式开发来说，PMON早期的版本扩展性不好。

目前，PMON已发展到PMON2000。PMON2000基于FreeBSD，支持MIPS、ARM、PPC和X86，并支持从flash、IDE、TFTP以及USB启动操作系统。同时，PMON2000内置调试系统，包含多种命令用于调试，能够运行应用程序，并使用串口作为输出。

1.1 PMON2000框架

对上述PMON2000目录(如图1所示)说明如下：

(1) Conf: Conf目录存放PMON2000的基本配置信息；

(2) Targets: 每个系统在该目录下有一个子目录，如GT64240、Bonito等。进行新系统支持的时候，需要修改该目录下的文件。以Bonito来为例子，主要有下列文件：

收稿日期: 2009-04-15; **修订日期**: 2009-08-25。

作者简介: 钱振江 (1982—)，男，江苏苏州人，硕士，助教，研究方向为软件工程、嵌入式系统； 常晋义 (1955—)，男，山西忻州人，教授，硕士生导师，研究方向为空间决策支持系统、软件工程； 张福新 (1976—)，男，博士，研究员，研究方向为微处理器系统结构设计、性能评估、机群计算。E-mail: zhenjiang.qian@gmail.com

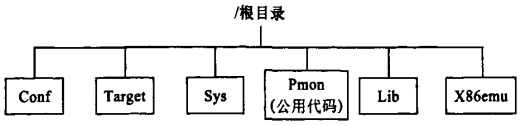

图 1 PMON2000 的目录框架

1) start.S：位于 Targets/Bonito/Bonito 目录下，是 C 环境建立之前的汇编代码，是整个 PMON 运行的起点；

2) tgt_machdep.c：位于 Targets/Bonito/Bonito 目录下，包括与各种开发板相关的函数；

3) pci_machdep.c：该文件包括 PCI 空间分配的实现函数；

4) Targets/Bonito/dev：该目录包含各种开发板所需的设备驱动；

5) Targets/Bonito/conf：该目录下是建立编译环境所需的各种配置文件。

(3) Sys：该目录存放系统支持文件。主要有以下子目录：

1) Arch：存放与处理器相关的代码定义；

2) Dev：该目录是设备的驱动程序，主要是 IDE 的 ATA、北桥的网卡 IC 以及 PCI 设备 eepro100 等；

3) Kern：主要是一些系统调用函数的具体实现，比如 malloc,time, signal, socket 函数等；

4) Net 和 netinet：该目录存放各种网络协议的实现文件；

5) Scsi：该目录是 SCSI 协议的实现文件。

(4) PMON(共有代码)：该目录存放的是 PMON 系统的公用代码，包括各种基本功能的实现，具体主要有以下子文件夹：

1) Arch：主要是与处理器相关的代码，比如 Flush_Cache 等；

2) Cmds：该目录存放 PMON2000 shell 各个命令的实现文件；

3) Dev：该目录存放基本设备的驱动，比如 Flash 等；

4) Fs：该目录是对文件系统的实现；

5) ELF 文件的 LOAD 方法的实现文件夹；

6) 各种网络命令以及 TFTP 功能的实现等。

(5) Lib：该目录是 PMON2000 的 lib 库，实现 memcpy、memset 以及 printf 等基本函数；

(6) X86emu：该目录是对 X86 的模拟(Emulator)，主要负责各种显卡的初始化和配置信息等。

现在龙芯系列平台采用的 PMON，在原来的 PMON 的基础上添加了硬盘支持、文件系统 ext2 和 fat 的支持，以及显卡的支持。同时，修复了 debug 功能，扩展性也得到提高，比较容易移植到新的系统。

1.2 PMON 命令配置

(1) Load 命令：用于加载文件。

语法为：load [-options] pathname。

其中 options 参数有：

-o offset：将文件加载到内存 load_address 加上 offset 处；

-f addr：将文件加载到 flash 的地址 addr 处；

Pathname：要加载的文件路径或 URL。

(2) g 命令：执行程序。

语法为：g [-st] [-b addr] [-e addr] [-- args]。

当 load 完一个文件后，通过 g 命令通知 PMON 开始执行刚刚载入的文件。

(3) devls 命令：显示设备。

(4) ifaddr 命令：配置网卡。

语法为：ifaddr ifname ipaddr[:ifparameters]。

(5) ping 命令：确定主机的网络连通性。

(6) reboot 命令：重新启动主机。

(7) set 命令：显示和设置各种环境参数。

(8) unset 命令：取消对各种环境参数的设置。

(9) setmac 命令：设置主机的 MAC 地址。

另外，包含的测试命令可以测试如 CPU 浮点部件、内存、网络、CPU 的频率、PCI 设备、显示、硬盘、键盘和串口等。

(10) usb 相关命令

usb help：列出所有 usb 命令；

usb stor：扫描所有 usb 存储设备；

usb info：显示所有可用的 usb 存储设备；

usb dev：显示当前所用之 usb 存储设备；

usb dev usb0：设置当前所用的 usb 存储设备为 usb0。

1.3 PMON2000 的空间分配

PMON2000 在内存中的空间分配方案如图 2 所示。

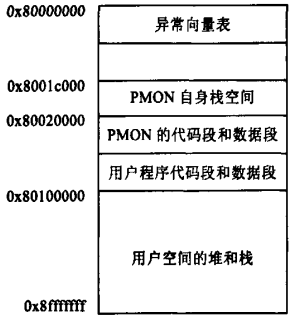

图 2 PMON2000 空间分配

2 关键技术

PMON2000 在龙芯 Mipsel 平台上的移植实现采用图 3 的流程。

PMON2000 开发实现分成两部分：汇编语言实现部分和 C 语言实现部分[6-7]。汇编语言实现部分主要包含各种硬件的基本初始化信息。由于后续程序可以在内存中运行，为此对南北桥芯片以及显卡等信息的初始化工作可以采用 C 语言实现。

2.1 Start. S 汇编程序部分

当龙芯主板加电后，CPU 将从 0xBFC00000 取指令开始执行[8]。并且 ROM 在系统中的起始地址也是 0xBFC00000，所以 ROM 中的第一条指令就是 CPU 运行的第一个指令。在 MIPS 中，异常处理的入口有两类，通过 CP0 的 STATUS 寄存器位 BEV 来决定：当 BEV=1 时，异常的入口地址为 0xBFC00000；而 BEV=0，异常的入口地址为 0x80000000。所以 PMON 程序段开始处是一些异常处理的入口，为此需要跳过这段空间，程序通过一个跳转 bal 指令跳转到该段后面开始执行，代码如下：

```
bal locate
nop
```

另外值得注意的是从 cache 空间到 uncache 空间的转换方

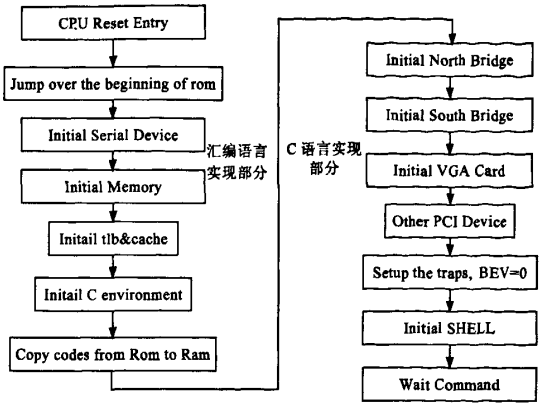

图 3　PMON2000 流程

法。转换的代码如下：

```
bal uncached
nop
bal locate
nop
uncached:
or ra, UNCACHED_MEMORY_ADDR
j      ra
nop
```

ra 中保留的是 bal locate 这条指令的地址，然后与 UNCACHED_MEMORY_ADDR 进行“或”操作，该地址就转换为 uncache 地址。

因为 Start.S 这段汇编程序在 Rom 执行，而编译后的数据段在 0x80020000 起始的空间，为了能够访问到数据段的数据，需要进行地址的修正，代码如下：

```
la s0, start
subu   s0, ra, s0
and s0, 0xffff0000
```

代码中的 s0 起到地址修正的目的。

接下来，开始进行 CPU 寄存器的初始化工作，包括清楚 TLB，并初始化北桥的基本信息，以确保 uart 能够正常工作；同时，初始化 uart，主要是设置波特率；然后初始化内存，主要通过 I2C 协议从内存的 EEPROM 读取内存参数来进行设置；最后，初始化 cache，复制 PMON 代码到内存，并执行以下代码：

```
la v0, initmips
jalr   v0
nop
```

至此程序开始在内存中运行。同时，由于可以读写内存，因此可以使用栈，所以接下来的步骤可以采用 C 语言实现。

2.2　C 语言部分

该部分在 Ram 即内存中运行，入口为 initmips 函数，该函数在文件 tgt_machdep.c 中实现，关键代码如下：

```
void initmips(unsigned int memsz){
    ...
    tgt_cpufreq();
    cpuinfotab[0] = &DBGREG;
    dbginit(NULL);
    ...
    main();
}
```

其中主要的初始化部分在 dbginit 函数中执行。

dbginit 函数中的各个关键函数说明如下：

(1) __init()：构造函数初始化，将所有的 constructor 函数执行一遍，建立一些基本的数据结构。在 PMON 中有 3 类 constructor 函数，它们都是静态函数，如下：

1)命令处理初始化函数，位于 pmon/cmds 目录下，其名称都叫 init_cmd()。init_cmd()是静态函数，几乎在每个命令文件中都有。

2)文件系统初始化函数，位于 pmon/fs 目录下，其名称叫 init_fs()或者类似形式，如 init_diskfs、init_fs、init_netfs 等每个都是静态函数。文件系统初始化也就是代表各个文件系统的数据结构插入到相应链表。对于磁盘文件系统，链表的头指针类型是 struct DiskFileSystem。对于其它的文件系统，头指针类型是 struct FileSystem。当需要对某个文件操作时，通过这两个链表可以找到相应的结构，再通过里面的函数指针就可以对文件进行具体操作了。

3)可执行文件类型初始化函数，该函数在 pmon/loaders 目录的名为 exec_*.c 的文件中，函数名为 init_exec()。

(2) envinit()：初始化环境变量；

(3) tgt_init()：初始化与板级相关的过程，在本系统中主要是初始化北桥和 PCI；

(4) inet_init()：初始化网络，并且间接地初始化显示接口。显示的初始化主要是通过调用 tgt_devconfig()函数来间接地调用 vga_bios_init()、radeon_init()、sm712_init()等函数实现的。

(5) hisinit()：初始化命令历史记录；

(6) ioctl(STDIN,TCGETA,&consterm)：建立终端。

对于整个 dbginit () 函数流程，在执行完以上几个函数以后，此时在 dbginit()中通过一个 switch 语句来根据用户的按键选择不同的执行流程。用于判断按键的函数为 get_boot_selection()，如果用户按下 Tab 键，则调用 recover()函数来进行系统恢复工作；如果用户按下'u'键，则调用 rescue()函数来从 U 盘上读取 boot.cfg 文件进行操作系统内核启动的工作；如果在规定的时间内没有按键或者按下 Enter 键，则通过调用 load_menu_list()函数来从硬盘上读取 boot.cfg 文件来进行操作系统内核的启动工作；如果用户按下 Del 键则进入 PMON console 来循环等待用户输入命令并且进行命令的解析执行工作。

值得注意的是，对于 CPU 频率的计算，可以使用 RTC 和 CPU count 寄存器来得到。利用 RTC 设定一段时间，看 count 在这段时间内的增加量，应为 count 的频率是总线频率一半。

2.3　PCI 的空间分配

PCI 的空间分配主要 3 个函数实现，调用关系依次为 tgt_devinit ()、_pci_businit () 和 _pci_hwinit ()。其中 _pci_hwinit () 为 PMON 中初始化 PCI 的主要函数，这个函数在 Target/Ev64240/pci/pci_machdep.c 中定义。

龙芯北桥有 3 个 64M 的 PCI 内存空间。256M~256M+

3*64M 是 CPU 访问 PCI 的内存空间；北桥中 PCIMAP 寄存器控制 3 个 PCI 内存窗口在整个 PCI 内存空间上的映射。在 PMON 中设置为 0x2040，也就是使用 0~192M 的 PCI 内存空间。

其中 pci_mem_base 为内存空间的基地址，pci_io_base 为 IO 空间的基地址。minipciioaddr 为 IO 空间的最小可以分配地址，minipcimemaddr 为内存空间的最小可以分配地址。nextpcimemaddr 为内存空间的下一个分配地址，nextpciioaddr 为 IO 空间的下一个分配地址。同时，在 PMON 中地址分配是逆序分配的。

值得一提的是 PCI 总线的扫描。北桥的初始化工作主要是建立设备可用的地址空间，在完成这样的初始化工作后，就开始进行 PCI 总线的扫描。_pci_scan_dev()函数通过 PCI 配置访问确定连接的设备，这通过树的深度遍历的算法实现，对于龙芯系统，因为只有一个 PCI 桥，因此树的深度是 2。

系统对于找到的每一个设备，查找相应的设备驱动，这是一个递归的过程。比如找到一个 USB 控制器，加载了控制器驱动之后，控制器驱动最后会扫描 USB 总线，并加载 USB 设备(如 usb storage、kbd)的驱动。

2.4 PMON 框架中 USB 模块的实现

USB 模块主要分为 USB 设备层、USB 中间协议层及 USB 主控制协议层，这些层次一层一层往下调，最终实现 USB 的功能。

对于上层的一些函数，如 usb_get_descriptor()等都是对应于 USB sepc 中提到的诸如对 USB 设备描述符进行获取等一系列标准的 USB 操作，这些操作都是在控制管道中进行的，用于对设备进行初始化、控制、获取状态等，当然最终都调用 usb_control_msg()来实现。usb_control_msg()根据传入的参数不同来通过不同的控制消息在控制管道中的传送，从而完成各种各样的对设备进行的控制操作。usb_control_msg()对 setup 包所需的数据结构进行填充以后继续调用本文件中的消息发送给 Root Hub。对于发送给 Root Hub 的控制消息，由于 Root Hub 是虚拟的，其所含信息都是静态写入代码的，所以控制消息可以马上送达，返回信息也马上返回，不存在延时的问题。

PMON USB 模块中大部分的 USB 操作请求最终都是对 submit_common_msg()的调用，为此说明 submit_common_msg()函数的实现。其整体框架主要是调用 sohci_submit_job()和 hc_check_ohci_controller()两个函数。sohci_submit_job()是真正实现提交消息给控制器的函数，包括 ED(endpoint descriptor)和 TD (transfer descriptor)等的分配与组装等。hc_check_ohci_controller()用于检测消息是否已经通过控制器送达给目标 USB 设备，然后做相应的事后处理工作。

2.5 PMON 编译环境的建立

根据龙芯系列处理器的特性完成对 PMON 系统文件的修改之后，需要对 PMON 进行编译，从而得到 PMON 的二进制文件。PMON 编译环境的建立过程如下：

(1) 将交叉编译工具包(如 comp.tar.gz)在自定义的目录(如/usr/local)利用 tar 等工具解开；

(2)将/usr/local/comp/mips-elf/gcc-2.95.3/bin 加入到 PATH 目录下，从而可以使用该交叉编译工具中的 gcc 等编译器；

(3) 进入 pmon2000 的 tools 目录下运行 make 预处理命令，建立一些 conf 需要的工具；

(4) 进入 pmon2000 的 Targets/Bonito/conf 目录中，编辑该目录下 Bonito 文件。在编译之前，可以选择需要编译的模块；

(5) 运行 tools/pmoncfg 目录下的 pmoncfg 命令，形成目标主目录下的 compiler 目录；

(6) 进入 Targets/Bonito/compiler/Bonito 的目录，再次运行 make 命令，从而编译生成二进制的 PMON 系统文件。

3 调 试

对于 PMON 进行调试采用以下分阶段的方法：

(1)在串口设备没有初始化前，利用逻辑分析仪进行测试。该环节的调试方法较为困难，所以应该尽早初始化串口。

(2)串口顺利工作后，可以利用串口通过调用 printf 函数输出各项参数来进行调试。

(3)初始化 PMON 系统的 shell 后，可以利用 PMON 系统中的调试系统进行测试。

4 结束语

本文在龙芯 Mipsel 平台上实现 PMON2000 的基本框架，为龙芯的各种应用平台提供类 BIOS 的支持。目前，该 PMON 版本在龙芯系列平台上运行稳定。基于稳定性的考虑，对 PMON 的整体结构没有进行大范围的修改，本文实现了 PMON 的底层基本框架，与目前主流 x86 平台上的 BIOS 存在着一定的差距，如图形化等。在以后的工作中，可以采用 GUI 来实现 PMON 的图形化系统，考虑到龙芯板级可用存储空间的限制，如何高效地实现图形化界面，同时又不会带来存储空间的急剧增加是一个重要的问题。在实现 PMON 基本功能的前提下，如何进一步裁剪系统也是后续工作需要解决的问题。

参考文献:

[1] Hu Wei-wu,Zhang Fu-xin,Li Zu-song. Micro-architecture of the Godson-2 processor[J].Journal of Computer Science and Technology,2005,20(2):243-249.

[2] 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.

[3] Dominic Sweetman.See MIPS run[M].2nd ed.北京:机械工业出版社,2007:13-17.

[4] Phil Bunce.PMON[OL].http://www.philbunce.com/pmon/,2002.

[5] 卡特索利斯.嵌入式硬件设计[M].徐君明,陈振林,郭天杰,编.2 版.北京:中国电力出版社,2007:121-138.

[6] 巴尔,马萨.嵌入式系统编程(英文影印版)[M]. 2 版.南京:东南大学出版社,2007:68-75.

[7] 诺尔加德.嵌入式系统硬件与软件架构[M].马洪兵,谷源涛,译.北京:人民邮电出版社,2008:34-45.

[8] 中国科学院计算技术研究所. 龙芯 2E 处理器数据手册 [Z]. 2006.