

**Q/LS**

**Q/LS 0046B-2021**

**龙芯中科技术股份有限公司企业标准**

**龙芯 CPU 统一系统架构规范**

**(适用于 LA 架构通用 PC、服务器系列)**

**V4.1**

2023-11-08 发布

2023-11-16 实施

**龙芯中科技术股份有限公司 批准**

## 版权声明

本档版权归龙芯中科技术股份有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此档中的任何部分公开、转载或以其他方式散发给第三方。否则，必将追究其法律责任。

## 免责声明

本档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因档使用不当造成的直接或间接损失，本公司不承担任何责任。

## 龙芯中科技术股份有限公司

Loongson Technology Corporation Limited

地址：北京市海淀区中关村环保科技示范园龙芯产业园 2 号楼

Building No.2, Loongson Industrial Park,

Zhongguancun Environmental Protection Park, Haidian District, Beijing

电话(Tel): 010-62546668

传真(Fax): 010-62600826

## 版本信息

版本记录	文档名	龙芯 CPU 统一系统架构规范
	版本号	V4.1
	创建人	研发部
版本历史		
序号	版本号	更新内容
1	V1.0	1、发布文档初始版本 V1.0 版
2	V1.2	1、更新《龙芯 3A4000_7A1000 硬件设计规范》为 V1.1 版； 2、新增《龙芯内置显卡软硬件设计规范》V0.2 版
3	V3.1	1、本规范版本更新为 V3.1 版； 2、更新《龙芯 3 号 CPU 固件与内核接口规范》为 V3.1 版； 3、更新《龙芯 3A5000_7A1000 固件开发规范》为 V0.3 版； 4、更新《龙芯 CPU 内核开发规范》为 V1.2 版； 5、更新《龙芯 7A1000 内置显卡软硬件设计规范》为 V1.0 版； 6、发布《龙芯 3A5000_7A1000 通用类板卡硬件设计规范》V1.1 版； 7、发布《龙芯 3A5000_7A1000 外围功能芯片支持列表》V1.0 版
4	V4.0	1、本规范版本更新为 V4.0 版； 2、更新《龙芯 3 号 CPU 固件与内核接口规范》为 V4.0 版； 3、更新《龙芯 7A1000 固件开发规范》为 V1.0 版； 4、发布《龙芯 7A2000 固件开发规范》V1.0 版； 5、发布《龙芯 3A5000 硬件设计规范》V1.0 版； 6、发布《龙芯 7A1000 硬件设计规范》V1.0 版； 7、发布《龙芯 7A2000 硬件设计规范》V1.0 版； 8、更新《龙芯 7A1000 内置显卡软硬件设计规范》为 V1.0 版； 9、发布《龙芯 7A2000 内置显卡软硬件设计规范》V1.1 版； 10、发布《龙芯外围功能芯片支持列表》V1.0 版
5	V4.1	1、本规范版本更新为 V4.1 版； 2、更新《龙芯 CPU 通用 PC/服务器系统固件与内核接口规范》为 V4.1 版； 3、新增《龙芯 7A 芯片组固件开发规范》V1.1 版； 4、新增《龙芯 2K2000 固件开发规范》V1.0 版； 5、更新《龙芯 7A2000 内置显卡软硬件设计规范》为 V1.2 版； 6、新增《龙芯 2K2000 内置显卡软硬件设计规范》V1.0 版； 7、新增《龙芯 LA 架构硬件设计规范》V1.0 版； 8、更新《龙芯外围功能芯片支持列表》为 V1.2 版；

# 1 序言

龙芯 CPU 统一系统架构规范是龙芯产品在桌面 PC、服务器领域的开发指导规范，通过规范整机设计、固件开发、内核开发的技术要求，定义了整机、固件、系统的兼容性标准。

## 2 发布列表

序号	规范名称	版本号
1	龙芯 CPU 通用 PC/服务器系统固件与内核接口规范	V4.1
2	龙芯 7A 芯片组固件开发规范	V1.1
3	龙芯 2K2000 固件开发规范	V1.0
4	龙芯 7A1000 内置显卡软硬件设计规范	V1.0
5	龙芯 7A2000 内置显卡软硬件设计规范	V1.2
6	龙芯 2K2000 内置显卡软硬件设计规范	V1.0
7	龙芯 LA 架构硬件设计规范	V1.0
8	龙芯外围功能芯片支持列表	V1.2

# 龙芯 CPU 通用 PC/服务器系统固件与内核 接口规范

**V4.1**

## 版本信息

文档信息	文档名	龙芯 CPU 通用 PC/服务器系统固件与内核接口规范
	版本号	V4.1
	创建人	系统研发部
版本历史		
序号	版本号	更新内容
1	V1.0	发布文档初始版本 V1.0 版。
2	V1.1	1、重新修订了地址规范约束； 2、重新修订了 SMBIOS 规范中 CPU 型号的约定；
3	V1.2	1、添加传参新成员 <code>cpuname</code> 及其格式规范； 2、添加新传参功能：表示桥片个数、DMA cache/uncache； 3、添加新的结构体描述（ <code>efi_reset_system_t</code> ）； 4、添加成员 <code>DoSuspend</code> ；
4	V2.0	1、重新修订附录 B <code>bootparam.h</code> 文件； 2、修正附录 A 中的几处错误；
5	V2.1	1、第 4 节修改，地址规范相关修改； 2、增加附录 E，添加 3A/3B+7A 描述（ <code>boardinfo</code> 、地址、中断），增加 7A GPU 使用地址段约束； 3、完善 4.2 节地址规范约束； 4、添加 4.3 节 DMA 规范描述以及 C、D、E 附录中 DMA 规范约束表； 5、修改一些编写错误；
6	V2.2	1、第 4.2 节，低端内存的高 16M 地址空间相对 UEFI 做了约束； 2、第 7 节，对 SMBIOS 产品信息（Type2）第六字段增加了命名约束，附录 A.6 <code>board_devices.name</code> 受同样约束； 3、附录 A.2，删除现在接口中已经废弃的 <code>screen_info</code> 和 <code>sys_desc_table</code> ； 4、附录 A.3，删除现在接口中已经废弃的 <code>sysstab</code> 和 UEFI runtime 服务相关的接口； 5、附录 A.4，规范了使用龙芯内置显卡情况 <code>vbios</code> 的处理； 6、附录 A.8，固件传参接口的内存映射表中对应地址空间范围增加了对应的 DMA 地址空间范围； 7、附录 A.9，固件传参接口添加 <code>of_dtb_addr</code> 成员； 8、附录 A.12，对固件传参接口 <code>cpuname</code> 的使用进行补充完善； 9、修改了一些语法错误和排版问题。
7	V2.3	1、4.2 节，低端内存搞 16M 地址空间 UEFI 参考分布更新； 2、4.3 节，DMA 地址规范删除，参考不同平台的固件设计规范；

		<p>3、5.1.2 节，修改中断处理方式，参考各平台固件开发规范；</p> <p>4、5.2 节，约束 LPC 接口使用方式；</p> <p>5、第 7 节，修改 SMBIOS 约定必须传递的信息以及实现参考；</p> <p>6、附录 A.4，更名 <code>smbios_tables</code> 为 <code>sysinfo_tables</code>；</p> <p>7、附录 A.7，修改 <code>vers</code> 成员的定义、根据实际使用情况修改其他成员定义；</p> <p>8、附录 B，更新 <code>bootparam.h</code> 文件；</p> <p>9、删除附录 C、D、E 各平台中断及地址空间约定，请分别参考对应平台的固件开发规范；</p> <p>10、修改了一些语法错误和排版问题。</p>
8	V3.0	<p>1、文档名称及正文中删除“开发系统”</p> <p>2、2 节，增加了约定章节和部分术语</p> <p>3、4.2 节，修改了地址空间的解释</p> <p>4、5.1.1 节，修改了固件内中断的分工</p> <p>5、6 节，改变了固件与内核的传参接口</p> <p>6、7 节，具体化了 <code>type2</code> 中 <code>product</code> 的格式约束</p> <p>7、8 节，新增了固件对 ACPI 规范的实现约束</p> <p>8、附录 A，更新为最新固件内核传参接口的的数据结构</p> <p>9、附录 B，增加 LINUX 操作系统键值表</p> <p>10、修改了一些语句描述和排版问题。</p>
9	V3.1	<p>1、8.5 节，FADT 的 <code>flags</code> 中新增支持 <code>PCI_EXP_WAK</code> 及 <code>RESET_REG_SUP</code> 标志使用描述。</p> <p>2、8.3 节、中断模型更新</p> <p>3、增加 8.8 节 MCFG 支持</p> <p>4、8.6.1 节，新增 <code>_SEG</code> 对象支持双桥</p> <p>5、6.3.1 节，3.0 接口 <code>bpi</code> 结构增加 64 位 <code>flags</code></p> <p>6、8.6.9 节，热键驱动增加 <code>VCBL</code> 的约束</p> <p>7、8.6.12 节，GPIO 多中断支持</p> <p>8、8.6.18 节，增加 ACPI 对 SE 设备的支持</p> <p>9、8.6.19 节，增加 ACPI 对温度传感器的支持</p> <p>10、8.9 节，增加 <code>SLIT</code> 表的支持</p> <p>11、8.10 节，增加 <code>SPCR</code> 表的支持</p>
10	V4.0	<p>1、修改了部分章节序号、图表序号以及语句描述</p> <p>2、2.1 节，添加了 <code>BootLoader</code> 的描述</p> <p>3、3 节，更新了架构关系</p>

		<p>4、6.1、6.2、6.3 和 6.4 节更新了传参约束</p> <p>5、更新了传参约定，删除 BPI 以及扩展链表相关内容</p> <p>6、8.4 节，修改了 SRAT 语句描述</p> <p>7、8.6.1，增加了 _OSC 方法的支持</p> <p>8、8.6.20 节，增加 EC 的支持</p> <p>9、8.6.21 节，增加 BMC 的支持</p> <p>10、9 节，增加了 VBIOS 的传递约定</p> <p>11、删掉 BPI 定义附录 A，添加了 ACPI 设备 ID 附录</p>
11	V4.1	<p>1、增加了 8.6.21 SDIO/MMC 配置</p> <p>2、增加了 8.6.22 I2S 音频配置</p> <p>3、10 节，修改了 screen_info EFI GUID</p> <p>4、增加了 8.11 PPTT 表</p> <p>5、增加了 8.6.23 SPI 配置</p> <p>6、更新了 8.6.12 GPIO 配置</p>



目录

1 范围.....	1
2 术语与约定.....	1
2.1 术语.....	1
2.2 约定.....	2
3 架构关系.....	2
4 地址空间规范.....	2
4.1 地址空间.....	2
4.2 DMA 地址映射规范.....	2
5 中断配置规范.....	3
5.1 配置方法.....	3
6 BootLoader 与内核传参规范.....	3
6.1 传参约定.....	3
6.2 命令行传递的约定.....	3
6.2.1 initrd.....	3
6.3 UEFI system table 传递的约定.....	4
6.3.1 memory map 表.....	4
6.3.2 initrd 表.....	4
6.4 其他约定.....	4
7 SMBIOS 规范的实现约定.....	5

8 ACPI 规范的实现约定.....	7
8.1 RSDP.....	7
8.2 XSDT.....	8
8.3 MADT.....	9
8.4 SRAT.....	12
8.5 FADT.....	13
8.6 DSDT.....	15
8.6.1 PCI 总线枚举.....	16
8.6.2 PCI 中断路由.....	17
8.6.3 设备电源管理.....	18
8.6.4 USB 设备配置.....	18
8.6.5 电池配置.....	18
8.6.6 电源适配器配置.....	22
8.6.7 处理器配置和控制.....	23
8.6.8 系统休眠唤醒.....	23
8.6.9 热键配置.....	23
8.6.10 热区管理.....	28
8.6.11 串口配置.....	28
8.6.12 GPIO 配置.....	29
8.6.13 I2C 配置.....	32
8.6.14 GPIO 模拟 I2C 配置.....	32
8.6.15 RTC 配置.....	32

8.6.16 PWM 配置.....	33
8.6.17 SE 设备.....	33
8.6.18 EC 设备.....	33
8.6.19 BMC 设备.....	34
8.6.20 DMA 配置.....	35
8.6.21 SDIO/MMC 配置.....	36
8.6.22 I2S 音频配置.....	36
8.6.23 SPI 配置.....	39
8.7 FACS.....	39
8.8 MCFG.....	39
8.9 SLIT.....	40
8.10 SPCR.....	40
8.11 PPTT.....	40
9 VBIOS 的传递约定.....	42
10 screen_info 传参约定.....	42
11 总结.....	43
附录 A LINUX 操作系统键值表.....	43
附录 B 龙芯 ACPI 设备 ID.....	56

# 1 范围

本规范规定了龙芯 CPU 的地址空间、中断配置、固件与内核接口传参实现、SMBIOS、ACPI 及 VBIOS 实现约定的要求。本规范适用于龙芯 LoongArch 架构系列 CPU。建议其它系统厂商遵循此规范开发相关产品。

本规范正文及附录 A、B 为通用规范，通用规范描述一般性的约定。针对不同的芯片组，请参阅对应的固件开发规范。

# 2 术语与约定

## 2.1 术语

本规范所用术语定义如下：

**固件：**Firmware，写入 ROM、EPROM 等非易失存储器中的程序，负责控制和协调集成电路。

**BIOS：**基本输入输出系统，Basic Input Output System，一组固化到主板上一个 ROM 芯片上的程序，它保存着计算机基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。BIOS 与硬件系统集成在一起，也被称为固件，本规范中固件和 BIOS 不做区分。

**UEFI：**统一的可扩展固定接口，Unified Extensible Firmware Interface，是 Intel 为全新类型的 PC 固件的体系结构、接口和服务提出的建议标准。主要目的是提供在 OS 加载之前在所有平台上一致、正确指定的启动服务，被看做是有近 20 多年历史的 PC BIOS 的继任者。

**UEFI system table：**包含了指向 UEFI runtime、boot services 和 Configuration Table 等内容的指针，具体可参考 UEFI 规范。

**UEFI Configuration table：**对应 UEFI system table 中的 ConfigurationTable 字段，包含了一个表数组，每个数组成员由 GUID 和对应表的指针构成，具体可参考 UEFI 规范。

**BootLoader：**引导加载程序，又称引导加载器、启动加载器或启动引导器，是由计算机固件（BIOS 或 UEFI）启动的软件，它负责用想要的内核参数加载内核，并根据配置文件初始化 RAM 磁盘。常见的 BootLoader 有：ELILO, SYSLINUX, EFI stub loader, GRUB 等

**PMON：**龙芯平台使用的一种兼有 BIOS 和 BootLoader 部分功能的开放源码软件。

**SMBIOS(System Management BIOS)：**是主板或系统制造者以标准格式显示产品管理信息所需遵循的统一规范。DMI (Desktop Management Interface)是帮助收集电脑系统信息的管理系统，DMI 信息的收集必须在严格遵照 SMBIOS 规范的前提下进行。SMBIOS 和 DMI 是由行业指导机构 Desktop Management Task Force (DMTF)起草的开放性的技术标准。

**HT(HyperTransport)：**是一种为主板上的集成电路互连而设计的端到端总线技术，目的是加快芯片间的数据传输速度。HT 通常指 CPU 到主板芯片（或北桥）之间的连接总线，即 HT 总线。类似于 Intel 平台中的前端总线（FSB），HT 按技术规格分有 HT1.0、HT2.0、HT3.0、HT3.1。

**PCI(Peripheral Component Interconnect)：**是连接电子计算机主板和外部设备的总线标准，用于定义局部总线的标准。此标准允许在计算机内安装多达 10 个遵从 PCI 标准的扩展卡。

**CPU(central processing unit)：**中央处理器，简称处理器。

**Core:** 处理器核，特指一个物理 cpu，是一个独立的硬件执行单元，有独立的寄存器和计算单元。

**Node:** 节点，NUMA 体系结构中的一个概念，一个 NUMA node 有一组 core 和内存，core 访问自身 node 内存（本地内存）的速度要快于访问其他 node 内存（远端内存）的速度，访问速度与 node 的距离有关。在没有特殊说明情况下，统一系统架构规范中提到的节点或 node，均为 NUMA 节点。

## 2.2 约定

(1) 本规范中的地址，未明确说明为虚拟地址时，均表示物理地址。

# 3 架构关系

龙芯 PC 产品的固件与内核接口在系统各软件之间所处的层次关系如图 3-1 所示：

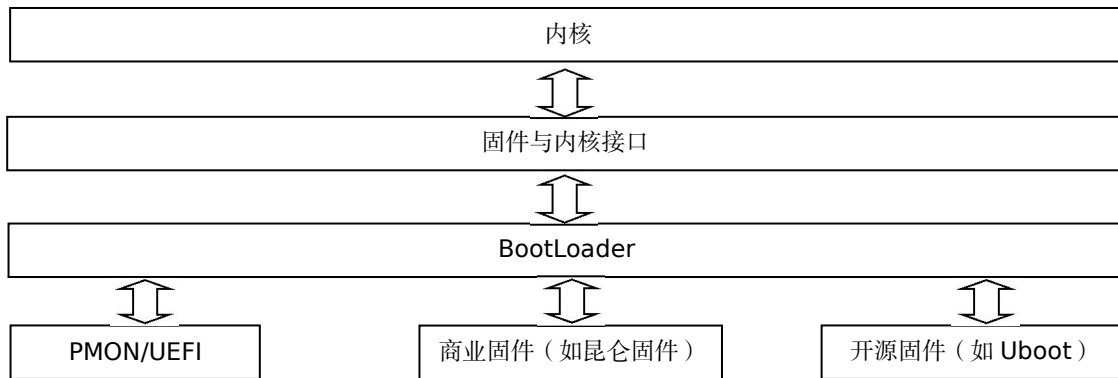


图 3-1 接口与内核和固件之间的关系

# 4 地址空间规范

## 4.1 地址空间

龙芯 CPU+ 芯片组的地址空间规定了内存空间、处理器及芯片组寄存器空间、PCI I/O 空间、PCI MEM 空间、PCI 配置空间，每种地址空间的范围，请参阅对应处理器、芯片组的固件开发规范。

## 4.2 DMA 地址映射规范

DMA 地址映射描述了系统中 DMA 地址与物理内存地址的转换关系，请参阅对应处理器、芯片组的固件开发规范。

## 5 中断配置规范

### 5.1 配置方法

根据 ACPI 规范，系统使用 GSI（global system interrupts，全局系统中断）为系统中断源分配中断号，并通过 ACPI 表实现中断配置，系统 GSI 分配详见对应处理器、芯片组的固件开发规范。

## 6 BootLoader 与内核传参规范

### 6.1 传参约定

BootLoader 传递给内核的参数包括标志、命令行地址、UEFI system table 地址，分别通过 a0、a1、a2 三个寄存器传递给内核。寄存器及对应的参数说明见下表：

表 6-1 传参寄存器及对应参数说明

寄存器名	参数大小	参数说明
a0	u64	UEFI 规范兼容标记, 为 1 1: 表示固件支持 UEFI 规范 0: 表示固件不支持 UEFI 规范 注: 当前规范版本仅支持 UEFI 兼容的固件, 固件不支持 runtime 时需将 runtime 指针置空, 并通过命令行的 noefi 禁止内核的 runtime 功能
a1	u64	64 位命令行物理地址
a2	u64	64 位 UEFI system table 物理地址

### 6.2 命令行传递的约定

#### 6.2.1 initrd

当使用 initrd 时，用户可通过命令行向内核传递 initrd 信息，initrd 信息的固定格式为“initrd=inird\_start,initrd\_size”，initrd\_start 为 initrd 的起始 64 位物理地址，initrd\_size 为 initrd 的 64 位大小。

## 6.3 UEFI system table 传递的约定

### 6.3.1 memory map 表

memory map 表的数据结构定义如下：

```
struct efi_boot_memmap {  
    unsigned long    map_size;  
    unsigned long    desc_size;  
    u32              desc_ver;  
    unsigned long    map_key;  
    unsigned long    buff_size;  
    efi_memory_desc_t    map[];  
};
```

其中，map\_size 为 UEFI memory map buffer 大小；desc\_size 为 UEFI memory map 描述符大小；desc\_ver 为 UEFI memory map 描述符版本；map\_key 为 UEFI memory map key；map 为 UEFI memory map 起始地址；具体请参考 UEFI 规范。buff\_size 保留。

memory map 表的地址存储在 UEFI Configuration table 中，内核在 UEFI Configuration table 中查找 memory map GUID({800f683f-d08b-423a-a293-965c3c6fe2b4})获取 memory map 地址。

### 6.3.2 initrd 表

initrd 表的数据结构定义如下：

```
struct efi_initrd {  
    unsigned long    base;  
    unsigned long    size;  
};
```

其中，base 为 initrd 的起始 64 位物理地址，size 为 initrd 的 64 位大小。

initrd 表的地址存储在 UEFI Configuration table 中，内核在 UEFI Configuration table 中查找 initrd GUID({5568e427-68fc-4f3d-ac74-ca555231cc68})获取 initrd 表地址。

## 6.4 其他约定

UEFI Configuration table 包含了 SMBIOS 表、ACPI 表的入口地址、自定义 screen\_info 表、memory map 表，详细描述参考章节 7 SMBIOS 规范的实现约定、章节 8 ACPI 的规范实现约定、章节 10 screen\_info 传参约定和章节 6.3.1 memory map 表。其中，传参涉及到的 SMBIOS 表、ACPI 表、screen\_info 表、memory map 表、initrd 的起始地址等地址均为 64 位物理地址且 64K 字节对齐。

initrd 有两种传递方式(分别对应 6.2.1 和 6.3.2 章节)，兼容 UEFI 规范的固件推荐使用 UEFI system table 传递。

## 7 SMBIOS 规范的实现约定

SMBIOS 是主板或系统制造者以标准格式显示产品管理信息所需遵循的统一规范。 DMI (Desktop Management Interface, DMI)就是帮助收集电脑系统信息的管理系统, DMI 信息的收集必须在严格遵照 SMBIOS 规范的前提下进行。SMBIOS 和 DMI 是由行业指导机构 Desktop Management Task Force (DMTF)起草的开放性的技术标准, 其中, DMI 设计适用于任何的平台和操作系统。 DMI 充当了管理工具和系统层之间接口的角色。它建立了标准的可管理系统更加方便了电脑厂商和用户对系统的了解。DMI 的主要组成部分是 Management Information Format (MIF)数据库。这个数据库包括了所有有关电脑系统和配件的信息。通过 DMI, 用户可以获取序列号、电脑厂商、串口信息以及其它系统配件信息。

SMBIOS 表的地址存储在 UEFI Configuration table 中, 内核在 UEFI Configuration table 中查找 SMBIOS GUID({EB9D2D31-2D88-11D3-9A16-0090273FC14D})获取 SMBIOS 地址。

龙芯固件平台必须实现的 **SMBIOS** 类别如下:

- BIOS 信息(Type 0)
- 系统信息 (Type 1)
- 产品信息 (Type 2)
- 系统外围或底架 (Type3)
- 处理器信息 (Type 4)
- 高速缓存信息 (Type 7)
- 系统插槽 (Type 9)
- 物理存储阵列 (Type 16)
- 存储设备(Type 17)
- 存储阵列映射信息 (Type 19)
- 表格结束指示 (Type 127)

其中:

**Type0**,描述固件信息。包括 BIOS 制造厂商、版本、ROM 大小等,其中, BIOS Characteristics Extension Byte 2 的 Bit 3 应与 6.1 章节 a0 一致。参考 SMBIOS3 系列协议实现。

**Type1**, 描述系统信息。通常指的是品牌整机的信息, 包括该机器的型号、版本、UUID 等信息, 参考 SMBIOS3 系列协议实现。

**Type2**, 描述主板信息。指主板制造商和主板本身信息, 包括主板生产厂家, 主板生产名称、版本、串号等, 参考 SMBIOS3 系列协议实现。

**Type3**, 系统外围或底架。《System Management BIOS (SMBIOS) Reference Specification》7.4 章节中规定了第六个字节的信息含义, 里面包含了主机的类型。见表 7-1。

表 7-1 SMBIOS Type3 第六字节信息含义

值	意义
01h	Other
02h	Unknown
03h	Desktop
04h	Low Profile Desktop
05h	Pizza Box



06h	Mini Tower
07h	Tower
08h	Portable
09h	Laptop
0Ah	Notebook
0Bh	Hand Held
0Ch	Docking Station
0Dh	All in One
0Eh	Sub Notebook
0Fh	Space-saving
10h	Lunch Box
11h	Main Server Chassis
12h	Expansion Chassis
13h	SubChassis
14h	Bus Expansion Chassis
15h	Peripheral Chassis
16h	RAID Chassis
17h	Rack Mount Chassis
18h	Sealed-case PC
19h	Multi-system chassis
1Ah	Compact PCI
1Bh	Advanced TCA
1Ch	Blade
1Dh	Blade Enclosure
1Eh	Tablet
1Fh	Convertible
20h	Detachable
21h	IoT Gateway
22h	Embedded PC
23h	Mini PC
24h	Stick PC

**Type4**, CPU 信息；描述 CPU 表项信息，由龙芯实现，无需固件和主板厂商填写，龙芯固件针对不同节点数量实现相应的参考代码。

注：ProcessorVersion 字段表示处理器名称；CurrentSpeed 字段表示处理器运行频率；CoreCount 字段表示一个封装内处理器核的数量。

**Type7**, Cache 信息；描述 Cache 组织结构信息。由龙芯实现，无需固件和主板厂商填写。

**Type9**, 系统插槽；描述主板的 PCI、PCIE 插槽的信息，该表项需要主板厂商根据各自设计自行完成；可参考龙芯固件代码实现。

**Type16**, 物理存储阵列；描述内存的信息，如大小、DIMM 槽数量、错误信息 Handle 等。该表项需要

主板厂商根据各自设计自行完成，可参考龙芯固件代码实现。

**Type17**，存储设备；描述每个内存槽的信息，比如类型、大小、是否有 ECC 等。该表项需要主板厂商根据各自设计自行完成，可参考龙芯固件代码实现。

**Type19**，存储阵列映射信息；描述内存映射到物理地址的范围。需要根据二级交叉开关的映射关系以及主存实际大小填写相应的地址范围；该表项需要固件和主板厂商根据自己 BIOS 地址映射关系进行填写，可参考龙芯固件代码实现。

**Type127**，表格结束标识；标识 SMBIOS 表的结尾，无需固件和主板厂商填写，龙芯代码已经实现。

## 8 ACPI 规范的实现约定

高级配置与电源接口（Advanced Configuration and Power Interface），简称 ACPI，是独立于体系结构的电源管理和配置框架，此框架建立了一个硬件寄存器集来定义电源状态（睡眠、休眠、唤醒等），并在软件上通过 ACPI 表的方式描述硬件信息、特性和控制特性的方法。ACPI 表列出了硬件板卡上无法使用硬件标准检测到或进行电源管理的设备，以及这些设备的功能。ACPI 表还列出了休眠电源状态、可用电源平面描述等系统功能。

本接口规范支持表项如 8-1：

表 8-1 支持表项

表	描述	是否强制
RSDP	Root System Description Pointer	是
XSDT	Extended System Description Table	是
MADT	Multiple APIC Description Table	是
SRAT	System Resource Affinity Table	是
FADT	Fixed ACPI Description Table	是
DSDT	Differentiated System Description Table	是
FACS	Firmware ACPI Control Structure	是
MCFG	PCI Express Memory-mapped Configuration Space base address description table	是
SLIT	System Locality Distance Information Table	否
SPCR	Serial Port Console Redirection Table	是
PPTT	Processor Properties Topology Table	否

注：建议同一厂家的各个表项的 OEM ID 保持相同

### 8.1 RSDP

RSDP（Root System Description Pointer）是整个 ACPI 表的第一个表，与其他表的关系可以用下图表示。

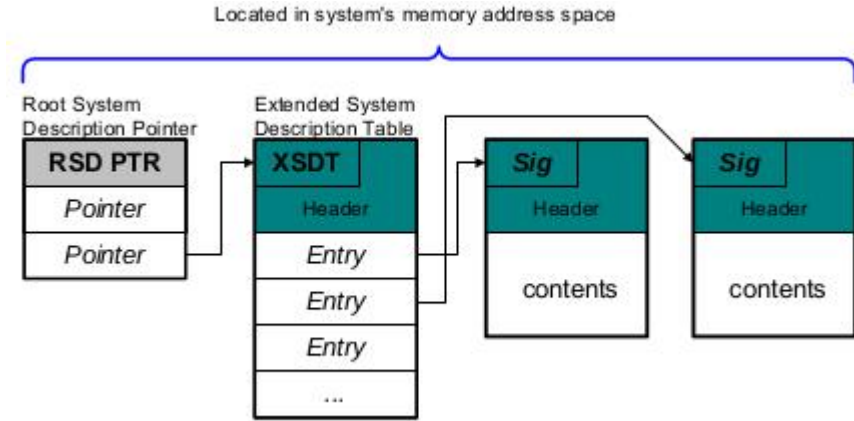


图 8-1 RSDP 示意图

RSDP 结构包含 XSDT（指向其他系统描述表的 64 位指针数组，见章节 8.2）的地址。RSDP 的地址存储 UEFI Configuration table 中，内核通过在 UEFI Configuration table 中查找 ACPI2.0 GUID ({8868e871-e4f1-11d3-bc22-0080c73c8881})来获取 RSDP 的地址。

RSDP 填写规范见表 8-2:

表 8-2 RSDP 结构

域	长度 (字节)	偏移 (字节)	描述
Signature	8	0	"RSD PTR"
Checksum	1	8	ACPI 1.0 规范中定义的字段的校验和。只包括该表的前 20 个字节，字节 0 到 19，包括校验和字段。这些字节的和必须为零。（代码实际实现）
OEMID	6	9	OEM 标识字符串
Revision	1	15	
RsdAddress	4	16	RSDT 的 32 位地址
Length	4	20	表的长度，包括表头，从偏移量 0 开始。用于记录整个表的大小。
XsdtAddress	8	24	XSDT 的 64 位地址
Extended Checksum	1	32	整个表的校验和，从偏移 0 开始到表结束，字节和为 0
Reserved	3	33	保留

## 8.2 XSDT

XSDT (Extended System Description Table) 包含其他系统描述表的 64 位指针数组，这些系统描述表向 OS 提供系统基本实现和配置的信息。部分定义如表 8-3:

表 8-3 XSDT 约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'XSDT'
Revision	1	8	1

## 8.3 MADT

MADT ( Multiple APIC Description Table ) 描述系统多核和中断信息。

表 8-4, 表列出了 MADT 部分约束条件, 未列出的部分参考 ACPI 规范。

LoongArch 架构支持的中断控制器模型包括 CORE PIC、LIO PIC、HT PIC、EIO PIC、MSI PIC、BIO PIC、LPC PIC 七种, 具体结构定义参考对应表项。

表 8-4 MADT 约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'APIC'
Revision	1	8	1
Local Interrupt Controller Address	4	36	处理器中断控制器寄存器基地址
Flags	4	40	0
Interrupt Controller Structure[n]		44	中断控制器结构列表

MADT 的 CORE PIC 结构参考如下配置:

表 8-5 CORE PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	CORE PIC 结构
Length	1	1	CORE PIC 结构字节长度
Version	1	2	版本号
ACPI Processor ID	4	3	处理器核 UID, 与 DSDT 处理器对象中的_UID 值相同
Physical Processor ID	4	7	CPU 核物理 ID
Flags	4	11	CORE PIC 的标志, 参考表 8-6

表 8-6 CORE PIC 标志

CORE PIC Flags	大小 (比特)	偏移 (比特)	描述
Enabled	1	0	0: CPU 不可用 1: CPU 可用

Reserved	31	1	必须为 0
----------	----	---	-------

MADT 的 LIO PIC 结构参考如下配置：

表 8-7 LIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	LIO PIC 结构类型
Length	1	1	LIO PIC 结构字节长度
Version	1	2	版本号
Base Address	8	3	LIO PIC 寄存器的基地址
Size	2	11	LIO PIC 寄存器空间大小
Cascade vector	2	13	描述了 LIO PIC 路由到 CORE PIC 的向量信息,每个字节代表一个 CORE PIC 向量号。
Cascade vector mapping	8	15	描述了路由到 CORE PIC 的 LIO PIC 向量信息, CORE PIC 向量由 Cascade vector 域指定,高 4 字节的 LIO PIC 向量对应于 Cascade vector 域高字节描述的 CORE PIC 向量,低 4 字节的 LIO PIC 向量对应于 Cascade vector 域低字节描述的 CORE PIC 向量。

MADT 的 HT PIC 结构参考如下配置：

表 8-8 HT PIC 结构

字段	大小(字节)	偏移(字节)	描述
Type	1	0	HT PIC 结构类型
Length	1	1	HT PIC 结构字节长度
Version	1	2	版本号
Base Address	8	3	HT PIC 寄存器的基地址
Size	2	11	HT PIC 寄存器空间大小
Cascade Vector	8	13	第 n 字节表示 $32*n \sim 31*(n+1)$ 的 HT PIC 中断向量路由到 LIO PIC 的中断向量

MADT 的 EIO PIC 结构参考如下配置：

表 8-9 EIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	EIO PIC 结构类型
Length	1	1	EIO PIC 结构字节长度
Version	1	2	版本号
Cascade vector	1	3	描述了 EIO PIC 路由到 CORE PIC 的 CORE PIC 向量号

Node	1	4	连接芯片组的处理器节点 ID
Node map	8	5	EIO 中断路由的处理器节点组，bit0-63 分别表示 0-63 节点

MADT 的 MSI PIC 结构参考如下配置：

表 8-10 MSI PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	MSI PIC 结构类型
Length	1	1	MSI PIC 结构字节长度
Version	1	2	版本号
Message Address	8	3	MSI 消息的目标地址
Start	4	11	MSI 在 HT PIC 或 EIO PIC 中的起始向量
Count	4	15	MSI 向量的个数

MADT 的 BIO PIC 结构参考如下配置：

表 8-11 BIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	BIO PIC 结构类型
Length	1	1	BIO PIC 结构字节长度
Version	1	2	版本号
Base Address	8	3	BIO PIC 寄存器的基地址
Size	2	11	BIO PIC 寄存器空间的大小
Hardware ID	2	13	BIO PIC 的硬件 ID，即 BIO PIC 所在芯片组连接的处理器节点的节点号
GSI base	2	15	BIO PIC 中断开始的 GSI 号，对于每个中断的 GSI， $GSI = GSI\ base + BIO\ PIC\ 的中断向量$

MADT 的 LPC PIC 结构参考如下配置：

表 8-12 LPC PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	LPC PIC 结构类型
Length	1	1	LPC PIC 结构字节长度
Version	1	2	版本号
Base Address	8	3	LPC PIC 寄存器的基地址
Size	2	11	LPC PIC 寄存器空间的大小
Cascade vector	2	13	描述了 LPC PIC 路由到 BIO PIC 的 BIO PIC 向量号

## 8.4 SRAT

SRAT ( System Resource Affinity Table ) 为操作系统提供了处理器和内存范围的亲和性关系，在 NUMA 平台上，操作系统启动期间依据此表进行配置。表头的约束见表 8-13。

龙芯平台需要实现两种亲和结构：

- (1) “Processor Local APIC/SAPIC Affinity Structure”，每个处理器核对应一个结构，见表 8-14；
- (2) “Memory Affinity Structure”，每个 Node 的连续内存段对应一个结构，见表 8-15。

表 8-13 SRAT 约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'SRAT'
Revision	1	8	2

表 8-14 Processor Local APIC/SAPIC Affinity 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0, 代表结构为 Processor Local APIC/SAPIC Affinity Structure。
Length	1	1	16
Proximity Domain [7:0]	1	2	处理器亲和域[7:0]
APIC ID	1	3	处理器 Local APIC ID, 见 MADT 表
Flags	4	4	标志。 0 位: 0 表示该 Processor Local APIC/SAPIC Affinity Structure 不可用; 1 表示该 Processor Local APIC/SAPIC Affinity Structure 可用。 [31:1]位: 必须为 0
Local SAPIC EID	1	8	用于 x86 架构的 SAPIC
Proximity Domain [31:8]	3	9	处理器亲和域[31:8]
Clock Domain	4	12	处理器时钟域

表 8-15 Memory Affinity 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	1, 代表结构为 Memory Affinity Structure
Length	1	1	40

Proximity Domain	4	2	内存亲和域
Reserved	2	6	保留
Base Address Low	4	8	内存范围地址的低 32 位
Base Address High	4	12	内存范围地址的高 32 位
Length Low	4	16	内存范围大小的低 32 位
Length High	4	20	内存范围大小的高 32 位
Reserved	4	24	保留
Flags	4	28	标志，指示内存区域是否已启用并可以热插拔，见表 8-16
Reserved	8	32	保留

表 8-16 内存热插拔标志

域	大小 (比特)	偏移 (比特)	描述
Enabled	1	0	0: 该 Memory Affinity Structure 不可用 1: 该 Memory Affinity Structure 可用
Hot Pluggable	1	1	是否支持内存热插拔
NonVolatile	1	2	是否为非易失内存
Reserved	29	3	0

## 8.5 FADT

FADT (Fixed ACPI Description Table) 为操作系统提供了 Fixed 硬件 ACPI 描述信息。

表 8-17 列出了 FADT 部分约束条件，未列出的部分请参考 ACPI 规范。

表 8-17 FADT 约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'FACP'
Length	4	4	表的长度，包括表头，从偏移量 0 开始。用于记录整个表的大小
FADT Major Version	1	8	FADT 版本号
FIRMWARE_CTRL	4	36	32 位 FACS 表地址
DSDT	4	40	32 位 DSDT 表地址
Reserved	1	44	0
SCI_INT	2	46	SCI 中断号
SMI_CMD	4	48	SMI 命令端口地址，龙芯平台无效



ACPI_ENABLE	1	52	写入 SMI_CMD 端口的数值，用于使能 SMI 对 ACPI 硬件寄存器的控制权
ACPI_DISABLE	1	53	写入 SMI_CMD 端口的数值，用于释放 SMI 对 ACPI 硬件寄存器的控制权
S4BIOS_REQ	1	54	写入 SMI_CMD 端口的数值，用于进入 S4BIOS 模式
PSTATE_CNT	1	55	非 0,则表示 OSPM 可向 SMI_CMD 写入该值，负责处理器性能控制
PM1a_EVT_BLK	4	56	PM1a Event Register Block 的地址
PM1b_EVT_BLK	4	60	PM1b Event Register Block 的地址
PM1a_CNT_BLK	4	64	PM1a Control Register Block 的地址
PM1b_CNT_BLK	4	68	PM1b Control Register Block 的地址
PM2_CNT_BLK	4	72	PM2 Control Register Block 的地址
PM_TMR_BLK	4	76	Power Management Timer Control Register Block 的地址
GPE0_BLK	4	80	General-Purpose Event 0 Register Block 的地址
GPE1_BLK	4	84	General-Purpose Event 1 Register Block 的地址
PM1_EVT_LEN	1	88	PM1a_EVT_BLK 长度，单位为字节
PM1_CNT_LEN	1	89	PM1a_CNT_BLK 长度，单位为字节
PM2_CNT_LEN	1	90	PM2_CNT_BLK 长度，单位为字节
PM_TMR_LEN	1	91	PM_TMR_BLK 长度，单位为字节
GPE0_BLK_LEN	1	92	GPE0_BLK 长度，单位为字节
GPE1_BLK_LEN	1	93	PM2_CNT_BLK 的长度
GPE1_BASE	1	94	ACPI GPE 模型中 GPE1 的_偏移量
CST_CNT	1	95	非 0,则 OSPM 可向 SMI_CMD 写入该值，表示支持_CST 对象及相应的通知
P_LVL2_LAT	2	96	大于 0x64 表示不支持 C2 状态
P_LVL3_LAT	2	98	大于 0x3e8 表示不支持 C3 状态
FLUSH_SIZE	2	100	一次刷 cache 的大小，单位字节，仅当 WBINVD 为 0 时有效
FLUSH_STRIDE	2	102	cache line 大小，单位字节，仅当 WBINVD 为 0 时有效
DUTY_OFFSET	1	104	P_CNT 中处理器的空闲周期起始值
DUTY_WIDTH	1	105	P_CNT 中处理器的空闲周期宽度
DAY_ALRM	1	106	RTC 日期的偏移地址
MON_ALRM	1	107	RTC 月份的偏移地址
CENTURY	1	108	RTC 世纪的偏移地址
IAPC_BOOT_ARCH	2	109	IA-PC 启动标志
Reserved	1	111	0
Flags	4	112	Fixed 特征标志，支持下列标志： WBINVD、PROC_C1、SLP_BUTTON、RESET_REG_SUP、 PCI_EXP_WAK 其中当 RESET_REG_SUP 为 1,表示通过内核操作 RESET_REG 复位，

			为 0 表示通过固件操作 RESET_REG 复位
RESET_REG	12	116	见表 8-18
RESET_VALUE	1	128	写入 RESET_REG 的值
ARM_BOOT_ARCH	2	129	ARM 平台启动标志
FADT Minor Version	1	131	0
X_FIRMWARE_CTRL	8	132	FACS 64bit 地址，固件动态生成
X_DSDT	8	140	DSDT 64bit 地址，固件动态生成
X_PM1a_EVT_BLK	12	148	PM1a Event Register Block 的 64 地址，见表 8-18
X_PM1b_EVT_BLK	12	160	PM1b Event Register Block 的 64 地址
X_PM1a_CNT_BLK	12	172	PM1a Control Register Block 的 64 位地址，见表 8-18
X_PM1b_CNT_BLK	12	184	PM1b Control Register Block 的 64 位地址
X_PM2_CNT_BLK	12	196	PM2 Control Register Block 的 64 位地址
X_PM_TMR_BLK	12	208	Power Management Timer Control Register Block 的 64 位地址，见表 8-18
X_GPE0_BLK	12	220	GPE0 block 的 64 位地址，见表 8-18
X_GPE1_BLK	12	232	GPE1 block 的 64 位地址

注：UEFI 固件动态生成 X\_FIRMWARE\_CTRL 与 X\_DSDT 之前，需将其初始化为 0。

表 8-18 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	代表为 System Memory
Reg Bit Width	1	1	寄存器位宽
Reg Bit Offset	1	2	寄存器偏移
Reserved	1	3	0
Address	8	4	寄存器地址

## 8.6 DSDT

DSDT (Differentiated System Description Table) 用来描述主板设计差异化信息。

本章节规定了龙芯平台 ACPI 设备 ID 和对 DSDT 配置的支持范围，支持范围之外的配置未验证，不保证其正确性。龙芯平台 ACPI 设备 ID 请参照附录 B。

以下各小节列出了龙芯平台支持的对象及方法，并对其中部分对象及方法进行了约束，未做约束的请参考 ACPI 规范。

## 8.6.1 PCI 总线枚举

支持的对象及方法：\_BBN、\_ADR、\_SEG、\_HID、\_CID、\_CRS、\_OSC。

(1) \_CRS (Current Resource Settings)

\_CRS 方法支持总线范围、IO 地址范围以及内存地址范围声明。

使用 WordBusNumber()宏声明 PCI 总线范围，各个参数规定如表 8-19：

表 8-19 WordBusNumber()宏参数表

参数名	值
ResourceUsage	描述总线范围的使用者是设备本身，或是下级总线设备
IsMinFixed	描述总线范围的最小值是否固定
IsMaxFixed	描述总线范围的最大值是否固定
Decode	描述设备解码总线范围的方向
AddressGranularity	总线范围起始号对齐
AddressMinimum	总线范围的最小值
AddressMaximum	总线范围的最大值
AddressTranslation	相对于主总线范围的偏移
RangeLength	总线范围大小
ResourceSourceIndex	资源描述符索引
ResourceSource	资源描述符
DescriptorName	资源描述符名称

使用 QWordIO()宏声明 PCI IO 地址范围，各个参数规定如表 8-20：

表 8-20 QWordIO()宏参数表

参数名	值
ResourceUsage	描述 I/O 范围的访问者
IsMinFixed	最小地址是否固定
IsMaxFixed	最大地址是否固定
Decode	描述 I/O 范围的访问方向
ISARanges	是否为 ISA 范围
AddressGranularity	地址对齐
AddressMinimum	起始地址
AddressMaximum	结束地址
AddressTranslation	相对于主总线的地址偏移
RangeLength	地址范围大小
ResourceSourceIndex	资源描述符索引

ResourceSource	资源描述符
DescriptorName	资源描述符名称
TranslationType	I/O 转换类型，下级总线是否与本总线资源相同，则不需要转换，否则需要指定 AddressTranslation
TranslationDensity	稀疏转换还是连续转换

使用 QWordMemory()宏声明 PCI MEMORY 地址范围，各个参数规定如表 8-21:

表 8-21 QWordMemory 参数表

参数名	值
ResourceUsage	描述内存范围的访问者
Decode	描述内存范围的访问方向
IsMinFixed	最小地址是否固定
IsMaxFixed	最大地址是否固定
Cacheable	是否支持 cache
ReadAndWrite	是否可读写
AddressGranularity	地址对齐
AddressMinimum	起始地址
AddressMaximum	结束地址
AddressTranslation	相对于主总线的地址偏移
RangeLength	地址范围大小
ResourceSourceIndex	资源描述符索引
ResourceSource	资源描述符
DescriptorName	资源描述符名称
MemoryRangeType	内存访问类型
TranslationType	转换类型，下级总线是否与本总线资源相同，则不需要转换，否则需要指定 AddressTranslation

## 8.6.2 PCI 中断路由

支持的对象及方法：\_PRT，描述芯片组 PCI 主桥上 PCI 中断分配信息。固件必须提供此对象，规定如下：

(1) \_PRT 中描述的 PCI 路由信息仅限于芯片组集成 PCI/PCIE 设备，对于外扩 PCI/PCIE 设备，无需中断配置。

(2) 表项中的中断号均使用 GSI 中断号，芯片组 GSI 分配详见芯片组固件开发规范。

(3) 对于芯片组的 PCIE PORT，\_PRT 中需要实现 4 个表项，分别对应 PCI 规范中规定的 A/B/C/D pin，4 个表项使用相同的 GSI 中断号。

(4) 对于芯片组的多功能设备，表项中的 pin 值为功能号。

### 8.6.3 设备电源管理

龙芯平台支持 USB2.0 和 GMAC 唤醒系统。

#### 8.6.3.1 USB

支持的对象及方法：\_PR0、\_PRW。

- (1) \_PR0 (Power State 0)

必须提供这个控制方法。

- (2) \_PRW (Power Resources for Wake)

仅支持 EventInfo 和 DeepestSleepState 参数。

#### 8.6.3.2 GMAC

支持的对象及方法：\_PR0、\_PRW。

- (1) \_PR0 (Power State 0)

必须提供这个控制方法。

- (2) \_PRW (Power Resources for Wake)

仅支持 EventInfo 和 DeepestSleepState 参数。

### 8.6.4 USB 设备配置

支持的对象及方法：\_UPC 及 \_PLD。

### 8.6.5 电池配置

支持的对象及方法：\_BIF \_BST \_STA，龙芯平台仅支持 Control Method Battery 电池模型。

按照 ACPI 规范要求，\_BIF 以及 \_BST 方法返回值为 Package，格式分别如表 8-22，8-23：

表 8-22 \_BIF 返回值 package 元素表

域	说明
Power Unit	电池信息使用单位
Design Capacity	电池设计容量
Last Full Charge Capacity	电池充满电时容量
Battery Technology	电池是否可充电
Design Voltage	电池设计电压
Design Capacity of Warning	电池告警电量
Design Capacity of Low	电池低电量

Battery Capacity Granularity 1	电池容量粒度 1（低电量到告警电量之间）
Battery Capacity Granularity 2	电池容量粒度 2（告警电量到满电量之间）
Model Number	电池型号
Serial Number	电池序列号
Battery Type	电池类型
OEM Information	OEM 特定信息

表 8-23 \_BST 返回值 package 元素表

域	说明
Battery State	电池状态：充电状态，放电状态，是否处于紧急状态
Battery Present Rate	电池当前功率或电流
Battery Remaining Capacity	电池剩余电量
Battery Present Voltage	电池当前电压

通过 EC（参见 8.6.20 节）管理电池时，\_BIF，\_BST，\_STA 方法的实现可以参照如下示例：

```

Device (EC)
{
    Name (_HID, Eisald ("PNP0C09") /* Embedded Controller Device */) // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
    Name (_GPE, 0x04) // _GPE: General Purpose Events
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        IO(Decode16, 0x62, 0x62, 0, 1,)
        IO(Decode16, 0x66, 0x66, 0, 1,)
    })
    OperationRegion (ECOR, EmbeddedControl, Zero, 0x0100)
    Field (ECOR, ByteAcc, NoLock, Preserve)
    {
        Offset (0x80),
        AST1, 1, //Battery Availability Status
        BCG1, 1, //Battery State
        LSTE, 1, //Lid Status
        RPWR, 1, //Power Source Status
        Offset (0x92),
        BVL, 8, //Battery Present Voltage Low 8 bits
        BVH, 8, //Battery Present Voltage High 8 bits
        BCL, 8, //Battery Present Rate Low 8 bits
        BCH, 8, //Battery Present Rate High 8 bits
        Offset (0x98),
        BRCL, 8, //Battery Remaining Capacity Low 8 bits
        BRCH, 8, //Battery Remaining Capacity High 8 bits
        BFCL, 8, //Battery Last Full Charge Capacity Low 8 bits
    }
}

```

```

BFCH,  8,    //Battery Last Full Charge Capacity High 8 bits
Offset (0xA2),
BDVL,  8,    //Battery Design Voltage Low 8 bits
BDVH,  8,    //Battery Design Voltage High 8 bits
Offset (0xAE),
BDCL,  8,    //Battery Design Capacity Low 8 bits
BDCH,  8,    //Battery Design Capacity High 8 bits
BSNL,  8,    //Battery Serial Number Low 8 bits
BSNH,  8,    //Battery Serial Number High 8 bits
}

Device (BAT0)
{
    Name (_HID, EisaId ("PNP0C0A") /* Control Method Battery */) // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
    Method (_STA, 0, NotSerialized) // _STA: Status
    {
        If (AST1)
        {
            Return (0x1F)
        }
        Else
        {
            Return (Zero)
        }
    }
}

Method (_BIF, 0, NotSerialized) // _BIF: Battery Information
{
    Name (BIFP, Package (0x0D))
    {
        One,
        0xFFFFFFFF,
        0xFFFFFFFF,
        One,
        0xFFFFFFFF,
        0xFFFFFFFF,
        0xFFFFFFFF,
        0xFFFFFFFF,
        0xFFFFFFFF,
        "LoongsonM",
        "LoongsonS",
        "LI-ION",
        ""
    }
}

```

```
    })
    Local1 = (BDVH * 0x0100)
    Local1 += BDVL
    BIFP [0x04] = Local1
    Local1 = (BDCH * 0x0100)
    Local1 += BDCL
    BIFP [One] = Local1
    Divide (Local1, 0x0A, Local2, BIFP [0x05])
    Divide (Local1, 0x14, Local2, BIFP [0x06])
    Divide (Local1, 0x64, Local2, BIFP [0x07])
    Divide (Local1, 0x64, Local2, BIFP [0x08])
    Local1 = (BFCH * 0x0100)
    Local1 += BFCL
    BIFP [0x02] = Local1
    Local1 = (BSNH * 0x0100)
    Local1 += BSNL
    BIFP [0x0A] = Local1
    Return (BIFP) /* \_SB_.PCI0.LPC_.EC_.BAT0._BIF.BIFP */
}

Method (_BST, 0, NotSerialized) // _BST: Battery Status
{
    Name (BSTP, Package (0x04)
    {
        0xFFFFFFFF,
        0xFFFFFFFF,
        0xFFFFFFFF,
        0xFFFFFFFF
    })
    If (RPWR)
    {
        If (BCG1)
        {
            BSTP [Zero] = 0x02
        }
        Else
        {
            BSTP [Zero] = Zero
        }
    }
    Else
    {
        BSTP [Zero] = One
    }
}
```



```
Local1 = (BCH * 0x0100)
BSTP [One] = (BCL + Local1)
Local1 = (BRCH * 0x0100)
BSTP [0x02] = (BRCL + Local1)
Local1 = (BVH * 0x0100)
BSTP [0x03] = (BVL + Local1)
Return (BSTP) /* _SB_.PCI0.LPC_.EC__.BAT0._BST.BSTP */
}
}
}
```

## 8.6.6 电源适配器配置

支持的对象及方法: `_PSR`。

通过 `EC` (参见 8.6.20 节) 管理电源适配器时, `_PSR` 方法的实现可以参照如下示例:

```
Device (EC)
{
    Name (_HID, Eisald ("PNP0C09") /* Embedded Controller Device */) // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
    Name (_GPE, 0x04) // _GPE: General Purpose Events
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        IO (Decode16, 0x0062, 0x0062, 0x01, 0x01,)
        IO (Decode16, 0x0066, 0x0066, 0x01, 0x01,)
    })
    OperationRegion (ECOR, EmbeddedControl, Zero, 0x0100)
    Field (ECOR, ByteAcc, NoLock, Preserve)
    {
        RPWR, 1 //Power Source Status
    }

    Device (AC)
    {
        Name (_HID, "ACPI0003" /* Power Source Device */) // _HID: Hardware ID
        Method (_PSR, 0, NotSerialized) // _PSR: Power Source
        {
            Return (RPWR) /* _SB_.PCI0.LPC_.EC__.RPWR */
        }
    }
}
```

## 8.6.7 处理器配置和控制

支持的对象及方法：\_HID、\_UID、\_PXM、\_STA、\_PPC、\_PCT、\_PSS。

### (1) \_PCT(Performance Control)

定义 ControlRegister 和 StatusRegister 时, AddressSpaceKeyword 必须为 FFixedHW, 其他域为任意值。

### (2) \_PSS (Performance Supported States)

- PSS 中的 Power 、 Bus master latency、 Status 为任意值。
- Latency: 大于 20000
- Control:
  - bits[31:9]: 保留;
  - bit[8]: 表示睿频标记, 1 表示睿频频率, 0 表示普频频率;
  - bits[7:0]: 表示频率等级, 范围 3-10。

## 8.6.8 系统休眠唤醒

支持的对象及方法：\_S0、\_S3、\_S4、\_S5 以及自定义的 S3 休眠地址。

S3 休眠地址是系统 S3 休眠时进入固件执行休眠的入口地址, 使用自定义对象声明, 规定如下:

- (1) 使用 Name 操作符定义的整数对象, 对象名称: SADR。
- (2) 该对象需要声明在 \_SB 下。

## 8.6.9 热键配置

龙芯平台提供了基于 EC (参见 8.6.20 节) 的统一热键驱动, 主板厂商可通过热键映射实现热键差异化设计, 规定如下:

- (1) 在 DSDT 中定义热键设备, HID 为 LOON0000。
- (2) 在热键设备中定义热键映射表, 名称为 “KMAP”, 每个表项代表一个热键, 表项格式约定如下: 表项类型为 package, 一个 package 由 3 个元素组成: 按键类型、按键索引、按键键值, 具体说明见表 8-24:

表 8-24 package 元素表

域	说明
---	----

按键类型	1 表示普通按键，操作系统键值表中前缀 KEY_ 的按键为普通按键 2 表示开关按键，操作系统键值表中前缀 SW_ 的按键为开关按键
按键索引	按键标识，大于 0 的任意整数，用于唯一标识一个按键，不同按键不能重复
按键键值	操作系统支持的键值码，参照附录 A

(1) 处理 EC 热键事件产生时，需要实现 EC 热键事件号 (EC 执行 QR\_EC 命令后，EC\_DATA 寄存器中存储的数值) 对应的查询方法。实现查询方法时，需要通过 Notify 通知热键驱动，通知对象为定义的热键设备对象，通知数据为一个无符号 16 位数据，格式约定如表 8-25:

表 8-25 通知数据格式

域	说明
bits[15:12]	按键类型: 1 表示普通按键 2 表示开关按键
bits[11:0]	按键索引

(2) 对于开关类型的热键，需要声明名称为 GSWS 的方法，返回开关的状态。状态数据为 32 位数据，每一位表示一个 SW 类型开关对应的开关状态，SW 类型开关的定义参考附录 A。

其中，GSWS 定义:

Method (GSWS, 0, NotSerialized)

作用: 获取开盖状态

参数: 无

返回值: 32 位正整数，开盖状态

通过 EC (参见 8.6.20 节) 管理开合盖时，可以参照如下示例:

```
Device (EC)
{
    Name (_HID, EisaId ("PNP0C09") /* Embedded Controller Device */) // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
    Name (_GPE, 0x04) // _GPE: General Purpose Events
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        IO (Decode16, 0x0062, 0x0062, 0x01, 0x01,)
        IO (Decode16, 0x0066, 0x0066, 0x01, 0x01,)
    })
    OperationRegion (ECOR, EmbeddedControl, Zero, 0x0100)
    Field (ECOR, ByteAcc, NoLock, Preserve)
    {
        LSTE, 1 //Lid Status
    }
    Name (ELID, Zero)
    Method (_Q21, 0, NotSerialized) // _Qxx: EC Query, xx=0x00-0xFF
    {
```

```
    ELID = One
    Notify (HKEY, 0x2007)
}

Device (HKEY)
{
    Name (_HID, "LOON0000") // _HID: Hardware ID
    Name (KMAP, Package (One)
    {
        Package (0x03)
        {
            0x02,
            0x07,
            Zero //SW_LID
        }
    })

    Method (GSWS, 0, NotSerialized)
    {
        If (ELID)
        {
            ELID = Zero
            Return (NOr (LSTE, 0xFFFFFFFF))
        }
        Return (Zero)
    }
}
```

### (1) 背光控制

热键驱动支持三种背光控制方式：

当热键设备定义 ECBS、ECBG、ECSL、ECLL、BLSW 方法时，表示通过热键驱动定义背光设备控制背光；

当热键驱动检测到 ACPI 背光设备，则调用热键设备中的 VCBL 方法并传参，参数为 false，表示由 ACPI 背光设备控制背光；

当热键驱动没有检测到 ACPI 背光设备，则由内置显卡背光设备控制背光。

其中，VCBL、ECBS、ECBG、ECSL、ECLL、BLSW 定义如下：

VCBL:

Method (VCBL, 1, Serialized)

作用：设置背光事件通知对象

参数：8 位正整数，0 表示背光事件应有 ACPI 背光设备接收

返回值：无

ECBS:

Method (ECBS, 1, Serialized)

作用: 设置背光亮度

参数: 16 位正整数, 亮度值

返回值: 无

ECBG:

Method (ECBG, 0, NotSerialized)

作用: 获取背光亮度

参数: 无

返回值: 16 位正整数, 亮度值

ECSL:

Method (ECSL, 0, NotSerialized)

作用: 获取支持的背光亮度最小值

参数: 无

返回值: 16 位正整数, 背光亮度最小值

ECLL:

Method (ECLL, 0, NotSerialized)

作用: 获取支持的背光亮度最大值

参数: 无

返回值: 16 位正整数, 背光亮度最大值

BLSW:

Method (\_BLSW, 1, Serialized)

作用: 控制系统背光开启/关闭

参数: 16 位正整数, 0 表示关, 1 表示开

返回值: 无

通过 EC (参见 8.6.20 节) 管理屏幕背光时, 可以参照如下示例:

```
Device (EC)
{
    Name (_HID, EisaId ("PNP0C09") /* Embedded Controller Device */) // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
    Name (_GPE, 0x04) // _GPE: General Purpose Events
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        IO (Decode16, 0x0062, 0x0062, 0x01, 0x01,)
```

```
IO (Decode16, 0x0066, 0x0066, 0x01, 0x01,)  
})  
OperationRegion (ECOR, EmbeddedControl, Zero, 0x0100)  
Field (ECOR, ByteAcc, NoLock, Preserve)  
{  
    BLAC,    8 //Backlight Adjust  
}  
  
Method (_Q38, 0, NotSerialized) // _Qxx: EC Query, xx=0x00-0xFF  
{  
    Notify (HKEY, 0x1005)  
}  
  
Method (_Q39, 0, NotSerialized) // _Qxx: EC Query, xx=0x00-0xFF  
{  
    Notify (HKEY, 0x1006)  
}  
  
Device (HKEY)  
{  
    Name (_HID, "LOON0000") // _HID: Hardware ID  
    Name (KMAP, Package (0x02))  
    {  
        Package (0x03)  
        {  
            One,  
            0x05,  
            0xE0 //KEY_BRIGHTNESSDOWN  
        },  
        Package (0x03)  
        {  
            One,  
            0x06,  
            0xE1 //KEY_BRIGHTNESSUP  
        }  
    }  
})  
Method (ECBS, 1, Serialized)  
{  
    BLAC = Arg0  
}  
  
Method (ECBG, 0, NotSerialized)  
{  
    Return (BLAC) /* \_SB_.PCI0.LPC_.EC_.BLAC */  
}
```

```

    }

    Method (ECSL, 0, NotSerialized)
    {
        Return (0x05)
    }

    Method (ECLL, 0, NotSerialized)
    {
        Return (0x64)
    }
}
}
}

```

### 8.6.10 热区管理

支持的对象及方法：\_CRT、\_PSL、\_PSV、\_TC1、\_TC2、\_TMP、\_TSP、\_TZP。

### 8.6.11 串口配置

支持的对象及方法：\_HID、\_UID、\_DSD、\_CRS。

#### (1) \_HID

龙芯平台为 PNP0501。

#### (2) \_UID

当使用处理器串口 0 时，\_UID 的值必须为 0，且此设备必须是 DSDT 中声明的第一个串口设备。

#### (3) \_CRS

\_CRS 方法仅支持寄存器和中断资源声明。

寄存器资源使用 QWordMemory()宏声明，传递寄存器地址信息，各个参数规定参考表 8-21。中断号资源使用 Interrupt()宏声明，各个参数的规定如表 8-26：

表 8-26 Interrupt()宏参数表

参数名	值
ResourceUsage	描述中断的使用者，该设备本身使用，或是子设备使用，为空，默认表示该设备本身使用
EdgeLevel	中断触发类型
ActiveLevel	中断触发极性
Shared	共享标志
ResourceSourceIndex	资源描述符索引
ResourceSource	资源描述符
DescriptorName	资源描述符名称
InterruptList	中断号

(4) \_DSD

\_DSD 对象仅支持串口时钟频率声明：

UUID：值为"daffd814-6eba-4d8c-8a91-bc9bbf4aa301"。

支持属性如表 8-27：

表 8-27 支持属性表

属性名	值	说明
clock-frequency	串口时钟频率	表示实际外接的串口时钟频率,单位为 Hz

### 8.6.12 GPIO 配置

支持的对象及方法：\_HID, \_UID, \_DSD, \_CRS。

(1) \_HID

7A1000 芯片集成的普通 GPIO 为 LOON0002, 3A5000 处理器集成的 GPIO 为 LOON0007, 2K2000 处理器集成的 GPIO 为 LOON000A, 2K2000 南北桥集成的 GPIO 为 LOON000B, 2K2000 南北桥集成的 ACPI GPIO 为 LOON000C, 7A2000 芯片集成的普通 GPIO 为 LOON000D, 7A2000 芯片集成的 ACPI GPIO 为 LOON000E, 3A6000 处理器集成的 GPIO 为 LOON000F。

(2) \_CRS

\_CRS 方法仅支持寄存器和中断资源声明。寄存器资源使用 QWordMemory()宏声明，传递寄存器地址信息，各个参数规定如表 8-21，中断号资源使用 Interrupt()宏声明，各个参数的规定如表 8-26。

(3) \_DSD

\_DSD 对象仅支持 GPIO 如下属性：

UUID：值为"daffd814-6eba-4d8c-8a91-bc9bbf4aa301"。

支持属性如表 8-28：

表 8-28 支持属性如表

属性名	说明
gpio_base	GPIO 在内核中的起始编号
ngpios	当前注册的 gpio 设备包含的 gpio pin 总数

注：具体配置约束见硬件设计规范章节

以 7A2000 ACPI GPIO 为例，DSDT 表可参照如下配置：

```

Device (GPO2)
{
    Name (_HID, "LOON000E") // _HID: Hardware ID
    Name (_ADR, Zero) // _ADR: Address
    Name (_UID, One) // _UID: Unique ID
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        QWordMemory (ResourceConsumer, PosDecode, MinFixed, MaxFixed, NonCacheable, ReadWrite,
            0x0000000000000000, // Granularity
            0x00000E00100D0080, // Range Minimum
    
```





```
}
Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive, ,, )
{
    0x0000006F,
}
Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive, ,, )
{
    0x0000006F,
}
Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive, ,, )
{
    0x0000006F,
}
Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive, ,, )
{
    0x0000006F,
}
Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive, ,, )
{
    0x0000006F,
}
Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive, ,, )
{
    0x0000006F,
}
})
Name (_DSD, Package (0x02) // _DSD: Device-Specific Data
{
    ToUUID ("daffd814-6eba-4d8c-8a91-bc9bbf4aa301") /* Device Properties for _DSD */,
    Package (0x02)
    {
        Package (0x02)
        {
            "gpio_base",
            0x70
        },
        Package (0x02)
        {
            "ngpios",
            16
        }
    }
})
}
```

### 8.6.13 I2C 配置

支持的对象及方法：\_HID，\_UID，\_CRS。

(1) \_HID

龙芯平台为 LOON0004。

(2) \_UID

必须为表示 I2C 总线号的任意正整数。

(3) \_CRS

\_CRS 方法仅支持寄存器资源声明。寄存器资源使用 QWordMemory()宏声明，传递寄存器地址信息，各个参数规定如表 8-21。

注：具体配置约束见硬件设计规范章节。

### 8.6.14 GPIO 模拟 I2C 配置

支持的对象及方法：\_HID，\_UID，\_DSD，\_CSR。

(1) \_HID

龙芯平台为 LOON0005。

(2) \_UID

必须为表示 I2C 总线号的任意正整数。

(3) \_DSD

\_DSD 对象仅支持如下属性：

UUID：值为"daffd814-6eba-4d8c-8a91-bc9bbf4aa301"。

支持属性如表 8-29：

表 8-29 支持属性表

属性名	说明
sda-gpio	指定 SDA 信号线使用的 GPIO 管脚
scl-gpio	指定 SCL 信号线使用的 GPIO 管脚
delay-us	总线传输时钟周期，单位为 us
timeout-ms	一次总线数据传输允许最大超时时间，单位为 ms，此为可选参数

### 8.6.15 RTC 配置

支持的对象及方法：\_HID，\_CRS。

(1) \_HID

龙芯平台为 LOON0001。

(2) \_CRS

\_CRS 方法仅支持寄存器和中断资源声明。寄存器资源使用 QWordMemory()宏声明，传递寄存器地址信息，各个参数规定如表 8-21，中断号资源使用 Interrupt()宏声明，各个参数的规定如表 8-26。

### 8.6.16 PWM 配置

支持的对象及方法：\_HID, \_UID, \_CRS。

(1) \_HID

龙芯平台为 LOON0006。

(2) \_CRS

\_CRS 方法仅支持寄存器和中断资源声明。寄存器资源使用 QWordMemory()宏声明，传递寄存器地址信息，各个参数规定如表 8-21，中断号资源使用 Interrupt()宏声明，各个参数的规定如表 8-26。

### 8.6.17 SE 设备

支持的对象及方法：\_HID、\_CRS。

(1) \_HID

龙芯平台为 LOON0003。

(2) \_CRS

\_CRS 方法仅支持中断资源声明。中断号资源使用 Interrupt()宏声明，各个参数的规定如表 8-26。

### 8.6.18 EC 设备

支持的对象及方法：\_CRS, HID, GPE, \_Qxx, \_DEP。

EC (Embedded Controller) 设备需要描述在 LPC 设备下，并通过 \_DEP 明确描述依赖的设备，可以通过 EC 管理电池，热键，键盘，鼠标等设备。管理电池可以参照 8.6.5 节中示例，热键管理及背光调节可以参照 8.6.9 节中示例，管理键盘，鼠标时可以参照如下示例：

```
Device (LPC)
{
    Name (_ADR, 0x00170000) // _ADR: Address
    Name (_S3D, 0x03) // _S3D: S3 Device State
    Name (RID, Zero)
    Device (EC)
    {
        Name (_HID, EisaId ("PNP0C09") /* Embedded Controller Device */) // _HID: Hardware ID
        Name (_UID, Zero) // _UID: Unique ID
        Name (_DEP, Package() {\SB.PCI0})
        Name (_GPE, 0x04) // _GPE: General Purpose Events
        Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
        {
            IO(Decode16, 0x62, 0x62, 0, 1,)
            IO(Decode16, 0x66, 0x66, 0, 1,)
        })
    }
}
```

```
Device (KBD)
{
    Method (_HID, 0, NotSerialized) // _HID: Hardware ID
    {
        Return (0x0303D041)
    }

    Name (_CID, Eisald ("PNP0303") /* IBM Enhanced Keyboard (101/102-key, PS/2 Mouse) */) //
_CID: Compatible ID
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        IO(Decode16, 0x60, 0x60, 1, 1,)
        IO(Decode16, 0x64, 0x64, 1, 1,)
        IRQ (Level, ActiveHigh, Exclusive, )
            {1}
    })
}

Device (MOU)
{
    Name (_HID, Eisald ("LEN2020")) // _HID: Hardware ID
    Name (_CID, Eisald ("PNP0F13") /* PS/2 Mouse */) // _CID: Compatible ID
    Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
    {
        IRQ (Level, ActiveHigh, Exclusive, )
            {12}
    })
}
}
}
```

## 8.6.19 BMC 设备

支持的对象及方法：\_STR、\_CRS，HID，\_IFT，\_SRV。

BMC 设备需要描述在 LPC 设备下。

## 8.6.20 DMA 配置

龙芯平台默认物理地址与 DMA 地址相同, 龙芯服务器为了兼容问题设备(能力寄存器标称支持 64 位 DMA 能力, 但实际上不支持 64 位 DMA 寻址能力, 比如仅支持 38 位 DMA 地址寻址能力), 固件可通过 `_DMA` 对象配置物理地址与 DMA 地址转换偏移。

以 2 节点为例子, 0 节点和 1 节点内存范围分别是 `0x20000 - 0x47ffffff`, `0x100080000000 - 0x10047ffffff`, 若问题设备最大可支持 38 位地址, 则需要将 1 位节点号放在 DMA 地址的 bit37, 上述物理地址范围对应的 DMA 地址范围分别是 `0x20000 - 0x47ffffff`, `0x2080000000 - 0x247ffffff`, 0 节点的节点号为 0, 所以物理地址与 DMA 地址相同, 1 节点物理地址于 DMA 地址的转换偏移为:

`0x100080000000-0x2080000000=0xfe00000000`, 相应的 `_DMA` 对象描述如下:

```
Name (_DMA, ResourceTemplate() {
    QWordMemory (ResourceProducer,
        PosDecode,
        MinFixed,
        MaxFixed,
        NonCacheable,
        ReadWrite,
        0x0,
        0x0000200000,
        0x047ffffff,
        0x0000000000,
        0x47fe00000,
        ,
        ,
    )
    QWordMemory (ResourceProducer,
        PosDecode,
        MinFixed,
        MaxFixed,
        NonCacheable,
        ReadWrite,
        0x0,
        0x100080000000,
        0x10047ffffff,
        0xfe00000000,
        0x400000000,
        ,
        ,
    )
})
```

## 8.6.21 SDIO/MMC 配置

支持的对象及方法: \_HID, \_UID, \_DSD, \_CSR。

### (1)\_HID

龙芯平台为 PRP0001。

### (2)\_DSD

\_DSD 对象仅支持如下属性:

UUID:值为"daffd814-6eba-4d8c-8a91-bc9bbf4aa301"。

compatible: linux 内核中 SDIO 驱动的匹配标识

dma-mask: 支持的 dma 位数

clock-frequency: 时钟频率

bus-width: 总线宽度

## 8.6.22 I2S 音频配置

I2S 声卡需要通过 I2S 控制器设备、Codec 设备和 Machine 设备实现配置，下面以 codec 下挂在芯片组 i2c1 控制器举例，配置实例如下：

```
Device (I2S0)
{
  Name (_ADR, 0x00070000) // _ADR: Address
  Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
  {
    Interrupt (ResourceConsumer, Level, ActiveHigh, Shared, ,, )
    {
      0x00000078,
      0x00000079,
    }
  })
  Name (_DSD, Package (0x02) // _DSD: Device-Specific Data
  {
    ToUUID ("daffd814-6eba-4d8c-8a91-bc9bbf4aa301") /* Device Properties for _DSD */,
    Package (0x02)
    {
      Package (0x02)
      {
        "clock-frequency",
        0x02FAF080
      },
      Package (0x02)
      {
```

```

        "interrupt-names",
        Package (0x02)
        {
            "tx",
            "rx"
        }
    }
}
}
})
}

Device (I2C1)
{
    Name (_HID, "LOON0004" /* AT Real-Time Clock */)
    Name (_UID, 0x1) // _UID: Unique ID
    Name (_CRS, ResourceTemplate ()
    {
        QWordMemory (ResourceConsumer, PosDecode, MinFixed, MaxFixed, NonCacheable, ReadWrite,
            0x0000000000000000,
            0x0000000010090100,
            0x0000000010090107,
            0x0000000000000000,
            0x0000000000000008,
            ,, , AddressRangeMemory, TypeStatic)
        Interrupt (ResourceConsumer, Level, ActiveHigh, Shared, ,, )
        {
            73,
        }
    })
}

Device(CODC)
{
    Name (_HID,"ESSX8323")
    Name (_CRS,ResourceTemplate()
    {
        I2cSerialBusV2 (0x0010, ControllerInitiated, 0x000186A0, AddressingMode7Bit,
            "\\_SB.I2C1", 0x00, ResourceConsumer, , Exclusive, )
    })
}
}

Device (PCA0)
{
    Name (_HID, "PRP0001") // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
}

```



```
Name (_DSD, Package (0x02) // _DSD: Device-Specific Data
{
  ToUUID ("daffd814-6eba-4d8c-8a91-bc9bbf4aa301") /* Device Properties for _DSD */,
  Package (0x05)
  {
    Package (0x02)
    {
      "compatible",
      "loongson,ls-audio-card"
    },

    Package (0x02)
    {
      "model",
      "Loongson-ASoC"
    },

    Package (0x02)
    {
      "mclk-fs",
      0x0100
    },

    Package (0x02)
    {
      "codec-name",
      "i2c-ESSX8326:00"
    },

    Package (0x02)
    {
      "codec-dai-name",
      "ES8326 HiFi"
    }
  }
})
}
```

同时，固件需要根据芯片手册配置正确的 I2S 管脚复用，并禁止 HDA 设备，以避免内核中 PCI 扫描阶段扫描到 HDA 设备。

## 8.6.23 SPI 配置

### (1)\_DSD

\_DSD 对象仅支持如下属性：

UUID:值为"daffd814-6eba-4d8c-8a91-bc9bbf4aa301"。

clock-frequency: 时钟频率

## 8.7 FACS

FACS（Firmware ACPI Control Structure）主要包含了唤醒向量地址，此表必须实现，其中部分约定如表 8-30，其他字段保留为 0。

表 8-30 FACS 表约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'FACS'
Firmware Waking Vector	4	12	32 位唤醒向量
Global Lock	4	16	全局锁
Flags	4	20	固件控制标志
X Firmware Waking Vector	8	24	64 位唤醒向量
Version	1	32	FACS 的版本

## 8.8 MCFG

MCFG（PCI Express Memory-mapped Configuration Space base address description table）主要包含了可访问到 PCIE 设备扩展配置空间基地址、PCI domain 域 ID 及总线范围。表头部分约定如表 8-31。PCIE 配置信息部分约定如表 8-32。

表 8-31 MCFG 约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'MCFG'
Revision	1	8	0x1

表 8-32 Memory Mapped Enhanced Configuration Space Base Address Allocation Structure 约束

域	大小 (字节)	偏移 (字节)	描述
Base Address	8	0	PCI 域扩展配置空间基地址

PCI Segment Group Number	2	8	PCI 域 ID, 注: 此值需与 DSDT 中 _SEG 对象的值一致
Start Bus Number	1	10	主桥下总线号最小值
End Bus Number	1	11	主桥下总线号最大值

## 8.9 SLIT

SLIT (System Locality Distance Information Table) 主要描述了系统不同节点间相对距离的信息, 此表为可选表项。表部分约束如表 3-33。

表 8-33 SLIT 约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'SLIT'
Number of System Localities	8	36	节点数
Entry[x][y]	1		当 $x = y$ , 相同节点相对距离 当 $x \neq y$ , 不同节点相对距离 xy 取值范围: $\leq$ Number of System Localities-1

## 8.10 SPCR

SPCR (Serial Port Console Redirection Table) 主要包含了串口配置, 此表选择实现, 其中部分约定如表 8-34。

表 8-34 SPCR 表部分约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'SPCR'
Interface Type	1	36	接口类型
Space ID	1	40	地址空间类型
Encoded Access Width	1	43	访问带宽编码
Address	8	44	基地址
Baud Rate	1	58	波特率

## 8.11 PPTT

PPTT (Processor Properties Topology Table) 主要描述了处理器的拓扑结构, 此表为可选表项, 其中表头的约束见表 8-35, Processor hierarchy node structure 约束见表 8-36。

表 8-35 PPTT 约束

域	大小	偏移	描述
---	----	----	----

	(字节)	(字节)	
Header			
Signature	4	0	'PPTT'
Revision	1	8	3

表 8-36 Processor hierarchy node structure 约束

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0 表示处理器结构
Length	1	1	本地处理器结构的长度，以字节为单位
Reserved	2	2	保留
Flags	4	4	处理器结构标志，见表 8-37
Parent	4	8	父处理器节点结构引用，值为 PPTT 表的开始和父处理器结构项的开始之间的差值。当没有父节点时，值为 0。
ACPI Processor ID	4	12	MADT 表中的处理器核 UID
Number of private resources	4	16	私有资源结构的个数
Private resources[N]	N*4	20	每个资源结构都表示处理器层次结构节点私有的资源

表 8-37 处理器结构标志

域	大小 (字节)	偏移 (字节)	描述
Physical package	1	0	如果处理器拓扑节点表示物理封装边界，则设置为 1； 如果处理器拓扑节点不表示物理封装边界，则设置为 0
ACPI Processor ID valid	1	1	该标志表示 ACPI Processor ID 是否指向有效的条目
Processor is a Thread	1	2	对于叶子表项，如果表示该处理器的处理元素与兄弟节点共享功能单元，则必须设置为 1； 对于非叶子表项，必须设置为 0
Node is a Leaf	1	3	如果 node 是处理器层次结构中的叶子节点，则必须设置为 1，否则必须设置为 0
Identical Implementation	1	4	值为 1 表示所有子处理器共享相同的实现
Reserved	2	5	0

## 9 VBIOS 的传递约定

对于内置显卡，固件需要将 VBIOS 存放到内置显卡 BAR2 空间的最后 1MB 内。

## 10 screen\_info 传参约定

screen\_info 为内核 framebuffer 设备提供信息，固件提供 EFI\_GRAPHICS\_OUTPUT\_PROTOCOL\_GUID(简称 GOP)时, BootLoader 通过 GOP 获取 screen\_info, 并注册自定义 screen\_info 表, 传递给内核。screen\_info 表 GUID: EFI\_GUID(0xe03fc20a, 0x85dc, 0x406e, 0xb9, 0x0e, 0x4a, 0xb5, 0x02, 0x37, 0x1d, 0x95)。

screen\_info 定义了显存基地址、分辨率、颜色位宽等, 和 Linux 内核 struct screen\_info 保持一致, 其数据结构定义如下表:

```
struct screen_info {
    UINT8  orig_x;           /* 0x00 */
    UINT8  orig_y;           /* 0x01 */
    UINT16 ext_mem_k;        /* 0x02 */
    UINT16 orig_video_page; /* 0x04 */
    UINT8  orig_video_mode; /* 0x06 */
    UINT8  orig_video_cols; /* 0x07 */
    UINT8  flags;           /* 0x08 */
    UINT8  unused2;         /* 0x09 */
    UINT16 orig_video_ega_bx; /* 0x0a */
    UINT16 unused3;         /* 0x0c */
    UINT8  orig_video_lines; /* 0x0e */
    UINT8  orig_video_isVGA; /* 0x0f */
    UINT16 orig_video_points; /* 0x10 */

    /* VESA graphic mode -- linear frame buffer */
    UINT16 lfb_width;        /* 0x12 */
    UINT16 lfb_height;       /* 0x14 */
    UINT16 lfb_depth;        /* 0x16 */
    UINT32 lfb_base;         /* 0x18 */
    UINT32 lfb_size;         /* 0x1c */
    UINT16 cl_magic, cl_offset; /* 0x20 */
    UINT16 lfb_linelength;   /* 0x24 */
    UINT8  red_size;         /* 0x26 */
    UINT8  red_pos;          /* 0x27 */
    UINT8  green_size;       /* 0x28 */
    UINT8  green_pos;        /* 0x29 */
    UINT8  blue_size;        /* 0x2a */
    UINT8  blue_pos;         /* 0x2b */
    UINT8  rsvd_size;        /* 0x2c */
}
```

```

UINT8  rsvd_pos;          /* 0x2d */
UINT16 vesapm_seg;       /* 0x2e */
UINT16 vesapm_off;      /* 0x30 */
UINT16 pages;           /* 0x32 */
UINT16 vesa_attributes; /* 0x34 */
UINT32 capabilities;    /* 0x36 */
UINT8  _reserved[6];    /* 0x3a */
};

```

## 11 总结

本传参规范的提出和制定旨在规范龙芯的固件和内核接口，重点规定了龙芯平台的地址空间划分、中断分配、传参数据结构、ACPI、VBIOS、SMBIOS 约定等方面，使得内核具有更广泛的适应性及兼容性，能有效解决内核对具体板卡设备的依赖性，有利于龙芯系列产品基础软件的规范和统一。

## 附录 A LINUX 操作系统键值表

按键名称	键值
KEY_RESERVED	0
KEY_ESC	1
KEY_1	2
KEY_2	3
KEY_3	4
KEY_4	5
KEY_5	6
KEY_6	7
KEY_7	8
KEY_8	9
KEY_9	10
KEY_0	11
KEY_MINUS	12
KEY_EQUAL	13
KEY_BACKSPACE	14
KEY_TAB	15
KEY_Q	16
KEY_W	17
KEY_E	18
KEY_R	19
KEY_T	20

KEY_Y	21
KEY_U	22
KEY_I	23
KEY_O	24
KEY_P	25
KEY_LEFTBRACE	26
KEY_RIGHTBRACE	27
KEY_ENTER	28
KEY_LEFTCTRL	29
KEY_A	30
KEY_S	31
KEY_D	32
KEY_F	33
KEY_G	34
KEY_H	35
KEY_J	36
KEY_K	37
KEY_L	38
KEY_SEMICOLON	39
KEY_APOSTROPHE	40
KEY_GRAVE	41
KEY_LEFTSHIFT	42
KEY_BACKSLASH	43
KEY_Z	44
KEY_X	45
KEY_C	46
KEY_V	47
KEY_B	48
KEY_N	49
KEY_M	50
KEY_COMMA	51
KEY_DOT	52
KEY_SLASH	53
KEY_RIGHTSHIFT	54
KEY_KPASTERISK	55

KEY_LEFTALT	56
KEY_SPACE	57
KEY_CAPSLOCK	58
KEY_F1	59
KEY_F2	60
KEY_F3	61
KEY_F4	62
KEY_F5	63
KEY_F6	64
KEY_F7	65
KEY_F8	66
KEY_F9	67
KEY_F10	68
KEY_NUMLOCK	69
KEY_SCROLLLOCK	70
KEY_KP7	71
KEY_KP8	72
KEY_KP9	73
KEY_KPMINUS	74
KEY_KP4	75
KEY_KP5	76
KEY_KP6	77
KEY_KPPLUS	78
KEY_KP1	79
KEY_KP2	80
KEY_KP3	81
KEY_KP0	82
KEY_KPDOT	83
KEY_ZENKAKUHANKAKU	85
KEY_102ND	86
KEY_F11	87
KEY_F12	88
KEY_RO	89
KEY_KATAKANA	90
KEY_HIRAGANA	91



KEY_HENKAN	92
KEY_KATAKANAHIRAGANA	93
KEY_MUHENKAN	94
KEY_KPJPCOMMA	95
KEY_KPENTER	96
KEY_RIGHTCTRL	97
KEY_KPSLASH	98
KEY_SYSRQ	99
KEY_RIGHTALT	100
KEY_LINEFEED	101
KEY_HOME	102
KEY_UP	103
KEY_PAGEUP	104
KEY_LEFT	105
KEY_RIGHT	106
KEY_END	107
KEY_DOWN	108
KEY_PAGEDOWN	109
KEY_INSERT	110
KEY_DELETE	111
KEY_MACRO	112
KEY_MUTE	113
KEY_VOLUMEDOWN	114
KEY_VOLUMEUP	115
KEY_POWER	116
KEY_KPEQUAL	117
KEY_KPPLUSMINUS	118
KEY_PAUSE	119
KEY_SCALE	120
KEY_KPCOMMA	121
KEY_HANGEUL	122
KEY_HANGUEL	KEY_HANGEUL
KEY_HANJA	123
KEY_YEN	124
KEY_LEFTMETA	125

KEY_RIGHTMETA	126
KEY_COMPOSE	127
KEY_STOP	128
KEY_AGAIN	129
KEY_PROPS	130
KEY_UNDO	131
KEY_FRONT	132
KEY_COPY	133
KEY_OPEN	134
KEY_PASTE	135
KEY_FIND	136
KEY_CUT	137
KEY_HELP	138
KEY_MENU	139
KEY_CALC	140
KEY_SETUP	141
KEY_SLEEP	142
KEY_WAKEUP	143
KEY_FILE	144
KEY_SENDFILE	145
KEY_DELETEFILE	146
KEY_XFER	147
KEY_PROG1	148
KEY_PROG2	149
KEY_WWW	150
KEY_MSDOS	151
KEY_COFFEE	152
KEY_SCREENLOCK	KEY_COFFEE
KEY_ROTATE_DISPLAY	153
KEY_DIRECTION	KEY_ROTATE_DISPLAY
KEY_CYCLEWINDOWS	154
KEY_MAIL	155
KEY_BOOKMARKS	156
KEY_COMPUTER	157
KEY_BACK	158

KEY_FORWARD	159
KEY_CLOSECD	160
KEY_EJECTCD	161
KEY_EJECTCLOSECD	162
KEY_NEXTSONG	163
KEY_PLAYPAUSE	164
KEY_PREVIOUSSONG	165
KEY_STOPCD	166
KEY_RECORD	167
KEY_REWIND	168
KEY_PHONE	169
KEY_ISO	170
KEY_CONFIG	171
KEY_HOMEPAGE	172
KEY_REFRESH	173
KEY_EXIT	174
KEY_MOVE	175
KEY_EDIT	176
KEY_SCROLLUP	177
KEY_SCROLLDOWN	178
KEY_KPLEFTPAREN	179
KEY_KPRIGHTPAREN	180
KEY_NEW	181
KEY_REDO	182
KEY_F13	183
KEY_F14	184
KEY_F15	185
KEY_F16	186
KEY_F17	187
KEY_F18	188
KEY_F19	189
KEY_F20	190
KEY_F21	191
KEY_F22	192
KEY_F23	193

KEY_F24	194
KEY_PLAYCD	200
KEY_PAUSECD	201
KEY_PROG3	202
KEY_PROG4	203
KEY_DASHBOARD	204
KEY_SUSPEND	205
KEY_CLOSE	206
KEY_PLAY	207
KEY_FASTFORWARD	208
KEY_BASSBOOST	209
KEY_PRINT	210
KEY_HP	211
KEY_CAMERA	212
KEY_SOUND	213
KEY_QUESTION	214
KEY_EMAIL	215
KEY_CHAT	216
KEY_SEARCH	217
KEY_CONNECT	218
KEY_FINANCE	219
KEY_SPORT	220
KEY_SHOP	221
KEY_ALTERASE	222
KEY_CANCEL	223
KEY_BRIGHTNESSDOWN	224
KEY_BRIGHTNESSUP	225
KEY_MEDIA	226
KEY_SWITCHVIDEOMODE	227
KEY_KBDILLUMTOGGLE	228
KEY_KBDILLUMDOWN	229
KEY_KBDILLUMUP	230
KEY_SEND	231
KEY_REPLY	232
KEY_FORWARDMAIL	233

KEY_SAVE	234
KEY_DOCUMENTS	235
KEY_BATTERY	236
KEY_BLUETOOTH	237
KEY_WLAN	238
KEY_UWB	239
KEY_UNKNOWN	240
KEY_VIDEO_NEXT	241
KEY_VIDEO_PREV	242
KEY_BRIGHTNESS_CYCLE	243
KEY_BRIGHTNESS_AUTO	244
KEY_BRIGHTNESS_ZERO	KEY_BRIGHTNESS_AUTO
KEY_DISPLAY_OFF	245
KEY_WWAN	246
KEY_WIMAX	KEY_WWAN
KEY_RFKILL	247
KEY_MICMUTE	248
KEY_OK	0x160
KEY_SELECT	0x161
KEY_GOTO	0x162
KEY_CLEAR	0x163
KEY_POWER2	0x164
KEY_OPTION	0x165
KEY_INFO	0x166
KEY_TIME	0x167
KEY_VENDOR	0x168
KEY_ARCHIVE	0x169
KEY_PROGRAM	0x16a
KEY_CHANNEL	0x16b
KEY_FAVORITES	0x16c
KEY_EPG	0x16d
KEY_PVR	0x16e
KEY_MHP	0x16f
KEY_LANGUAGE	0x170
KEY_TITLE	0x171

KEY_SUBTITLE	0x172
KEY_ANGLE	0x173
KEY_ZOOM	0x174
KEY_MODE	0x175
KEY_KEYBOARD	0x176
KEY_SCREEN	0x177
KEY_PC	0x178
KEY_TV	0x179
KEY_TV2	0x17a
KEY_VCR	0x17b
KEY_VCR2	0x17c
KEY_SAT	0x17d
KEY_SAT2	0x17e
KEY_CD	0x17f
KEY_TAPE	0x180
KEY_RADIO	0x181
KEY_TUNER	0x182
KEY_PLAYER	0x183
KEY_TEXT	0x184
KEY_DVD	0x185
KEY_AUX	0x186
KEY_MP3	0x187
KEY_AUDIO	0x188
KEY_VIDEO	0x189
KEY_DIRECTORY	0x18a
KEY_LIST	0x18b
KEY_MEMO	0x18c
KEY_CALENDAR	0x18d
KEY_RED	0x18e
KEY_GREEN	0x18f
KEY_YELLOW	0x190
KEY_BLUE	0x191
KEY_CHANNELUP	0x192
KEY_CHANNELDOWN	0x193
KEY_FIRST	0x194

KEY_LAST	0x195
KEY_AB	0x196
KEY_NEXT	0x197
KEY_RESTART	0x198
KEY_SLOW	0x199
KEY_SHUFFLE	0x19a
KEY_BREAK	0x19b
KEY_PREVIOUS	0x19c
KEY_DIGITS	0x19d
KEY_TEEN	0x19e
KEY_TWEN	0x19f
KEY_VIDEOPHONE	0x1a0
KEY_GAMES	0x1a1
KEY_ZOOMIN	0x1a2
KEY_ZOOMOUT	0x1a3
KEY_ZOOMRESET	0x1a4
KEY_WORDPROCESSOR	0x1a5
KEY_EDITOR	0x1a6
KEY_SPREADSHEET	0x1a7
KEY_GRAPHICSEDITOR	0x1a8
KEY_PRESENTATION	0x1a9
KEY_DATABASE	0x1aa
KEY_NEWS	0x1ab
KEY_VOICEMAIL	0x1ac
KEY_ADDRESSBOOK	0x1ad
KEY_MESSENGER	0x1ae
KEY_DISPLAYTOGGLE	0x1af
KEY_BRIGHTNESS_TOGGLE	KEY_DISPLAYTOGGLE
KEY_SPELLCHECK	0x1b0
KEY_LOGOFF	0x1b1
KEY_DOLLAR	0x1b2
KEY_EURO	0x1b3
KEY_FRAMEBACK	0x1b4
KEY_FRAMEFORWARD	0x1b5
KEY_CONTEXT_MENU	0x1b6

KEY_MEDIA_REPEAT	0x1b7
KEY_10CHANNELSUP	0x1b8
KEY_10CHANNELSDOWN	0x1b9
KEY_IMAGES	0x1ba
KEY_DEL_EOL	0x1c0
KEY_DEL_EOS	0x1c1
KEY_INS_LINE	0x1c2
KEY_DEL_LINE	0x1c3
KEY_FN	0x1d0
KEY_FN_ESC	0x1d1
KEY_FN_F1	0x1d2
KEY_FN_F2	0x1d3
KEY_FN_F3	0x1d4
KEY_FN_F4	0x1d5
KEY_FN_F5	0x1d6
KEY_FN_F6	0x1d7
KEY_FN_F7	0x1d8
KEY_FN_F8	0x1d9
KEY_FN_F9	0x1da
KEY_FN_F10	0x1db
KEY_FN_F11	0x1dc
KEY_FN_F12	0x1dd
KEY_FN_1	0x1de
KEY_FN_2	0x1df
KEY_FN_D	0x1e0
KEY_FN_E	0x1e1
KEY_FN_F	0x1e2
KEY_FN_S	0x1e3
KEY_FN_B	0x1e4
KEY_BRL_DOT1	0x1f1
KEY_BRL_DOT2	0x1f2
KEY_BRL_DOT3	0x1f3
KEY_BRL_DOT4	0x1f4
KEY_BRL_DOT5	0x1f5
KEY_BRL_DOT6	0x1f6



KEY_BRL_DOT7	0x1f7
KEY_BRL_DOT8	0x1f8
KEY_BRL_DOT9	0x1f9
KEY_BRL_DOT10	0x1fa
KEY_NUMERIC_0	0x200
KEY_NUMERIC_1	0x201
KEY_NUMERIC_2	0x202
KEY_NUMERIC_3	0x203
KEY_NUMERIC_4	0x204
KEY_NUMERIC_5	0x205
KEY_NUMERIC_6	0x206
KEY_NUMERIC_7	0x207
KEY_NUMERIC_8	0x208
KEY_NUMERIC_9	0x209
KEY_NUMERIC_STAR	0x20a
KEY_NUMERIC_POUND	0x20b
KEY_NUMERIC_A	0x20c
KEY_NUMERIC_B	0x20d
KEY_NUMERIC_C	0x20e
KEY_NUMERIC_D	0x20f
KEY_CAMERA_FOCUS	0x210
KEY_WPS_BUTTON	0x211
KEY_TOUCHPAD_TOGGLE	0x212
KEY_TOUCHPAD_ON	0x213
KEY_TOUCHPAD_OFF	0x214
KEY_CAMERA_ZOOMIN	0x215
KEY_CAMERA_ZOOMOUT	0x216
KEY_CAMERA_UP	0x217
KEY_CAMERA_DOWN	0x218
KEY_CAMERA_LEFT	0x219
KEY_CAMERA_RIGHT	0x21a
KEY_ATTENDANT_ON	0x21b
KEY_ATTENDANT_OFF	0x21c
KEY_ATTENDANT_TOGGLE	0x21d
KEY_LIGHTS_TOGGLE	0x21e

KEY_ALS_TOGGLE	0x230
KEY_ROTATE_LOCK_TOGGLE	0x231
KEY_BUTTONCONFIG	0x240
KEY_TASKMANAGER	0x241
KEY_JOURNAL	0x242
KEY_CONTROLPANEL	0x243
KEY_APPSELECT	0x244
KEY_SCREENSAVER	0x245
KEY_VOICECOMMAND	0x246
KEY_ASSISTANT	0x247
KEY_BRIGHTNESS_MIN	0x250
KEY_BRIGHTNESS_MAX	0x251
KEY_KBDINPUTASSIST_PREV	0x260
KEY_KBDINPUTASSIST_NEXT	0x261
KEY_KBDINPUTASSIST_PREVGROUP	0x262
KEY_KBDINPUTASSIST_NEXTGROUP	0x263
KEY_KBDINPUTASSIST_ACCEPT	0x264
KEY_KBDINPUTASSIST_CANCEL	0x265
KEY_RIGHT_UP	0x266
KEY_RIGHT_DOWN	0x267
KEY_LEFT_UP	0x268
KEY_LEFT_DOWN	0x269
KEY_ROOT_MENU	0x26a
KEY_MEDIA_TOP_MENU	0x26b
KEY_NUMERIC_11	0x26c
KEY_NUMERIC_12	0x26d
KEY_AUDIO_DESC	0x26e
KEY_3D_MODE	0x26f
KEY_NEXT_FAVORITE	0x270
KEY_STOP_RECORD	0x271
KEY_PAUSE_RECORD	0x272
KEY_VOD	0x273

KEY_UNMUTE	0x274
KEY_FASTREVERSE	0x275
KEY_SLOWREVERSE	0x276
KEY_DATA	0x277
KEY_ONSCREEN_KEYBOARD	0x278
SW_LID	0x00
SW_TABLET_MODE	0x01
SW_HEADPHONE_INSERT	0x02
SW_RFKILL_ALL	0x03
SW_RADIO	SW_RFKILL_ALL
SW_MICROPHONE_INSERT	0x04
SW_DOCK	0x05
SW_LINEOUT_INSERT	0x06
SW_JACK_PHYSICAL_INSERT	0x07
SW_VIDEOOUT_INSERT	0x08
SW_CAMERA_LENS_COVER	0x09
SW_KEYPAD_SLIDE	0x0a
SW_FRONT_PROXIMITY	0x0b
SW_ROTATE_LOCK	0x0c
SW_LINEIN_INSERT	0x0d
SW_MUTE_DEVICE	0x0e
SW_PEN_INSERTED	0x0f
SW_MAX	0x0f

## 附录 B 龙芯 ACPI 设备 ID

设备 ID	说明
LOON0000	热键设备
LOON0001	芯片组 RTC 模块
LOON0002	7A1000 芯片组集成的普通 GPIO
LOON0003	SE 模块
LOON0004	芯片组 I2C 控制器
LOON0005	GPIO 模拟 I2C
LOON0006	芯片组 PWM 控制器
LOON0007	3A5000 处理器集成的 GPIO
LOON0008	处理器温度传感器

LOON0009	芯片组 CAN 控制器
LOON000A	2K2000 处理器集成的 GPIO
LOON000B	2K2000 南北桥集成的 GPIO
LOON000C	2K2000 南北桥集成的 ACPI GPIO
LOON000D	7A2000 芯片组集成的普通 GPIO
LOON000E	7A2000 芯片组集成的 ACPI GPIO
LOON000F	3A6000 处理器集成的 GPIO

# 龙芯 7A 芯片组固件开发规范

**V1.1**

## 版本信息

文档信息	文档名	龙芯 7A 芯片组固件开发规范
	版本号	V1.1
	创建人	系统研发部
版本历史		
序号	版本号	更新内容
1	V0.1	发布文档初始版本 V0.1 版。
2	V0.2	1 增加对 ACPI GSI 的支持。 2 增加对 irqdomain 的支持。 3 增加对 ACPI 配置参考信息。
3	V0.3	1 FADT 的 flags 更新 2 中断模型更新 3 MCFG 支持 4 多桥支持 5 SLIT 支持 6 GPIO 多中断支持 7 SE 设备 ACPI 支持 8 温度传感器配置支持 9 SPCR 支持 10 背光调节支持
4	V1.0	1、修改了部分章节序号、图表序号以及语句描述 2、1.1 节更新配置寄存器基址 3、删除了原 1.2 节 LPC 控制器的约束 4、1.2 节更新了 VBIOS 传递方式 5、1.3 节更新了 GMAC 的约束 6、1.5 节增加了多芯片组的中断相关描述 7、2.3.1 节增加了 _OSC 方法的支持 8、2.3.11 节增加了 EC 配置 9、2.5 更新了 S3 休眠地址的对象数值 10、2.3 去掉温度传感器配置
5	V1.1	1、1.6.1 节修改了地址空间的部分地址约定 2、2.3.1 节修改了 64 位 PCI MEM 地址范围 3、增加了 1.1.2 I2S 约束 4、增加了 2.3.12 节 SPI 配置 5、增加了 2.10 节 PPTT 配置 6、2.3.5 节修改了 GPIO 配置

目录

前 言.....	1
1 设计约束.....	2
1.1 Audio.....	2
1.1.1 HDA.....	2
1.1.2 I2S.....	2
1.2 VBIOS.....	2
1.3 GMAC.....	2
1.4 DMA 地址映射.....	3
1.5 中断.....	3
1.5.1 龙芯 5000 系列处理器+7A 芯片组中断.....	3
1.6 地址空间.....	7
1.6.1 龙芯 5000 系列处理器+7A 芯片组地址空间.....	7
2 参考配置.....	9
2.1 MADT 配置.....	9
2.2 FADT 配置.....	11
2.3 DSDT 配置.....	14
2.3.1 PCI 总线枚举.....	14
2.3.2 设备电源管理.....	21
2.3.3 系统休眠唤醒.....	21
2.3.4 串口配置.....	22
2.3.5 GPIO 配置.....	23

2.3.6 I2C 配置.....	25
2.3.7 RTC 配置.....	26
2.3.8 PWM 配置.....	27
2.3.9 热区管理.....	28
2.3.10 SE 设备.....	31
2.3.11 EC 配置.....	31
2.4 FACS 配置.....	31
2.5 S3 休眠地址.....	32
2.6 SRAT 配置.....	32
2.7 MCFG 配置.....	33
2.8 SLIT 配置.....	34
2.9 SPCR 配置.....	35
2.10 PPTT 配置.....	35



## 前 言

本规范面向固件软件开发人员，在《龙芯 CPU 通用 PC/服务器系统固件与内核接口规范 V4.1》基础上针对 7A 芯片组的固件开发约束进行补充说明。

## 1 设计约束

### 1.1 Audio

#### 1.1.1 HDA

固件通过 Verb Table 机制实现不同板卡中对声卡的差异化设计。固件开发人员需要根据具体主板的声卡设计正确填写 Verb Table。具体参考龙芯代码中关于 Realtek 的 Verb Table 配置实现。

注意：

须确认芯片组 HDA 引脚工作在 HDA 模式（芯片组配置寄存器偏移 0x0440 寄存器的 12:11 为 x1b。芯片组配置寄存器基址需要固件下进行配置，参考代码配置寄存器基址为：0x0e0010010000）。

#### 1.1.2 I2S

对于 7A2000，固件需要根据芯片手册配置 hda\_i2s\_sel 引脚为 I2S 工作模式（芯片组配置寄存器偏移 0x0440 寄存器的 12:11 为 10b，配置寄存器基址需要固件下进行配置，参考代码配置寄存器基址为：0x0e0010010000）。

### 1.2 VBIOS

7A 芯片组内部包含图形处理器（以下简称 GPU），位于 D6:F0；内核下 GPU 需要 Video BIOS（以下简称 VBIOS）的支撑才可以工作正常。

固件对 VBIOS 处理方式：将固件内默认 VBIOS 全部读取到内置显卡 BAR2 空间的最后 1MB。

### 1.3 GMAC

7A1000 包含两个 GMAC 控制器，固件分别从 7A1000 SPI flash 偏移 0x0 和 0x10 处读取 MAC0 和 MAC1 地址信息，并判断是否合法，如果合法，并写入相应 GMAC 的 MAC 地址寄存器中；如果非法，将随机数写入相应 GMAC 的 MAC 地址寄存器中。

7A2000 包含一个 GMAC 控制器，固件从 7A2000 SPI flash 偏移 0x0 处读取 MAC0 地址信息，并判断是否合法，如果合法，写入相应 GMAC 的 MAC 地址寄存器中；如果非法，将随机数写入相应 GMAC 的 MAC 地址寄存器中。

MAC 地址寄存器定义见表 1-1 和 1-2：

表 1-1 GMAC0/GMAC1 Address High Register(Offset 0x40)

位域	名称	默认值	描述
31	MO: Always 1	0x0	保留

30:1 6	Reserved	0x0	保留
15:0	MAC Address0[47:32] MAC 地址高 16 位	0x0	存放用于接收地址过滤和传输流控帧的 MAC 地址。

表 1-2 GMAC0/GMAC1 Address Low Register(Offset 0x44)

位域	名称	默认值	描述
31:0	MAC Address0[31:0] MAC 地址低 32 位	0x0	存放用于接收地址过滤和传输流控帧的 MAC 地址。

## 1.4 DMA 地址映射

对于不支持超过 44 位 DMA 地址的设备，固件需要调整 7A 芯片组的 DMA 路由配置以支持此类设备。参考对应 7A 芯片组用户手册，DMA 路由配置寄存器（0x041C）的[12:8]位，具体实现参考龙芯代码。寄存器定义见表 1-3：

表 1-3 DMA 路由配置寄存器[12:8]位

位域	名称	访问	描述
12:8	dma_node_id_offset	RW	DMA 访问中节点号所在的地址偏移（相对于 bit36）

## 1.5 中断

### 1.5.1 龙芯 5000 系列处理器+7A 芯片组中断

龙芯 5000 系列处理器+7A 芯片组方案，芯片组中断控制器管理的中断可通过中断线或 HT 路由至处理器，MSI 中断通过 HT 路由至处理器，以上中断最终路由至处理器核。见图 1-1 所示。

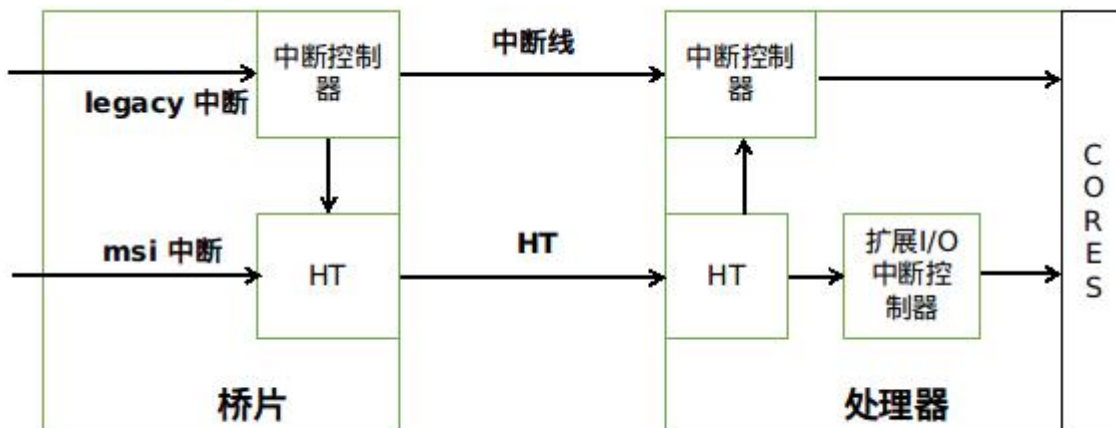


图 1-1 龙芯 5000 系列处理器+7A 芯片组中断路由示意图

7A 芯片组系统依据 ACPI 规范使用 GSI ( global system interrupts, 全局系统中断) 为中断源分配虚拟化的中断号, 一个 GSI 类型的虚拟中断号唯一标识了其所属的中断域, 内核按照不同的中断域实现中断管理。各个中断域对应的 GSI 中断号范围约定如下:

- 0-15: LPC/ISA 中断域, 用于系统外扩 LPC 设备中断源, 如表 1-4 所示, 具体分配情况参考 LPC 设备说明。

表 1-4 LPC/ISA 中断域 GSI 分配

GSI 中断号	中断源	说明
0	HPET	HPET 高精度定时器
1	I8042	键盘
2	级联	
3		
4		
5		
6		
7		
8	RTC	实时时钟
9		
10		
11		
12	I8042	鼠标
13		
14	ide0	硬盘
15	Ide1	硬盘

- 16-47: 处理器 I/O 中断控制器中断域, 其 GSI 中断号与中断源向量号的映射关系:  
GSI 中断号 = 中断源向量号 + 16  
中断源向量号为处理器手册中某个中断源在中断控制寄存器中的索引, 具体参考龙芯 5000 系列处理器手册。

- 48-63: 保留。

- 64-127: 芯片组 I/O 中断控制器中断域。其 GSI 中断号与中断源向量号的映射关系:  
GSI 中断号 = 中断源向量号 + GSI base  
中断源向量号为芯片组手册中某个中断源在中断控制寄存器中的索引, 具体参考对应龙芯 7A 芯片组用户手册;

GSI base 为 MADT 表中当前芯片组 BIO PIC 结构的 GSI base 成员。

其中单芯片组集成或外接的 PCI 兼容设备分配的中断号如表 1-5、1-6:

表 1-5 7A1000 单芯片组 I/O 中断控制器中断域 GSI 分配

中断号	中断源	中断号	中断源
64		96	pcie_f0_0
65		97	pcie_f0_1
66		98	pcie_f0_2

67		99	pcie_f0_3
68		100	pcie_f1_0
69		101	pcie_f1_1
70		102	pcie_h_lo
71		103	pcie_h_hi
72	uart[3:0]	104	pcie_g0_hi
73	i2c[5:0]	105	pcie_g0_lo
74		106	pcie_g1_lo
75		107	pcie_g1_hi
76	gmac0_sbd	108	toy[0]
77	gmac0_pmt	109	toy[1]
78	gmac1_sbd	110	toy[2]
79	gmac1_pmt	111	acpi
80	sata[0]	112	usb_0_ehci
81	sata[1]	113	usb_0_ohci
82	sata[2]	114	usb_1_ehci
83	lpc	115	usb_1_ohci
84		116	rtc[0]
85		117	rtc[1]
86		118	rtc[2]
87		119	hpet_int
88	pwm[0]	120	ac97_dma[0]
89	pwm[1]	121	ac97_dma[1]
90	pwm[2]	122	ac97/hda
91	pwm[3]	123	gpio_hi
92	dc	124	gpio[0]
93	gpu	125	gpio[1]
94	gmem	126	gpio[2]
95	thsens	127	gpio[3]

表 1-6 7A2000 单芯片组 I/O 中断控制器中断域 GSI 分配

中断号	中断源	中断号	中断源
64		96	pcie_f0_0
65		97	pcie_f0_1
66		98	pcie_f0_2
67	rio	99	pcie_f0_3
68	prg	100	pcie_f1_0
69	sataphy	101	pcie_f1_1

70	usb3phy	102	pcie_h_lo
71	gmac0_phy	103	pcie_h_hi
72	uart[3:0]	104	pcie_g0_lo
73	i2c[5:0]	105	pcie_g0_hi
74	vpu	106	pcie_g1_lo
75	can	107	pcie_g1_hi
76	gmac0_sbd	108	toy[0]
77	gmac0_pmt	109	toy[1]
78	gmac1_sbd	110	toy[2]
79	gmac1_pmt	111	acpi
80	sata	112	usb_0_ehci
81		113	usb_0_ohci
82		114	usb_1_ehci
83	lpc	115	usb_1_ohci
84	hpet[1]	116	rtc[0]
85	hpet[2]	117	rtc[1]
86	usb3	118	rtc[2]
87	hda1	119	hpet[0]
88	pwm[0]	120	i2s_dma[0]
89	pwm[1]	121	i2s_dma[1]
90	pwm[2]	122	i2s/hda
91	pwm[3]	123	gpio_hi
92	dc	124	gpio[0]/gpio[13]
93	gpu	125	gpio[1]/gpio[14]
94	gmem	126	gpio[2]/gpio[15]
95	thsens	127	gpio[3]/gpio[50]

注意：

(1) 根据 7A1000 硬件规范，若使用 EC 时需要使用 SCI 中断，此中断需要硬件连接到 7A1000 电源管理模块的 LID 管脚，信号约束参照硬件设计规范，对应 GSI 中断源为 111 号。

(2) 根据 7A2000 硬件规范，若使用 EC 时需要使用 SCI 中断，此中断需要硬件连接到 7A2000 电源管理模块的 ACPI GPIO 管脚，信号约束参照硬件设计规范，对应 GSI 中断源为 111 号。

(3) 对于多芯片组系统,非 PCI 兼容设备的中断源仅在一个芯片组上可用。

## 1.6 地址空间

### 1.6.1 龙芯 5000 系列处理器+7A 芯片组地址空间

龙芯 5000 系列处理器+7A 芯片组地址空间如图 1-2 所示：

0xYF FF FFFF FFFF	Reserved
0xYF 00 0000 0000	PCI EXT CFG
0xYE FE 0000 0000	Reserved
0xYE FD 0000 0000	PCI EXT MEM
0xYE 80 0000 0000	7A HT1 lo
0xYE 00 0000 0000	Reserved
MEM_MAX_ADDR	System RAM
0xY0 00 9000 0000	System RAM/Reserved
0xY0 00 8000 0000	PCI MEM
0xY0 00 3000 0000	CPU Registers
0xY0 00 1C00 0000	PCI CFG
0xY0 00 1A00 0000	PCI I/O
0xY0 00 1800 0000	Chipset Registers
0xY0 00 1000 0000	System RAM/Reserved
0xY0 00 0000 0000	System RAM/Reserved

图 1-2 龙芯 5000 系列处理器+7A 芯片组地址空间划分

地址中的 Y 代表节点号（0-15），各个地址空间说明如下：

0xY0 00 0000 0000 - 0xY0 00 0FFF FFFF：对于 0 号节点，表示内存，对于非 0 号节点，表示保留空间

0xY0 00 1000 0000 - 0xY0 00 17FF FFFF：连接到处理器节点 Y 的芯片组寄存器

0xY0 00 1800 0000 - 0xY0 00 19FF FFFF：连接到处理器节点 Y 的 PCI I/O 空间

0xY0 00 1A00 0000 - 0xY0 00 1BFF FFFF：连接到处理器节点 Y 的 PCI 配置空间

0xY0 00 1C00 0000 - 0xY0 00 1FFF FFFF：处理器节点 Y 的低速设备寄存器

0xY0 00 3000 0000 - 0xY0 00 7FFF FFFF：连接到处理器节点 Y 的 32 位 PCI MEM 空间

0xY0 00 8000 0000 - 0xY0 00 8FFF FFFF：对于非 0 号节点，表示内存，对于 0 号节点，表示保留空间

0xY0 00 9000 0000 - MEM\_MAX\_ADDR：内存，MEM\_MAX\_ADDR 的取值见表 1-7：

表 1-7 MEM\_MAX\_ADDR 的取值

系统中的节点数	MEM_MAX_ADDR
1	0xY0 80 0000 0000
2	0xY0 80 0000 0000
4	0xY0 40 0000 0000

8	0xY0 20 0000 0000
16	0xY0 10 0000 0000

0xY1 00 0000 0000 - 0xYD FF FFFF FFFF: 保留  
 0xYE 00 0000 0000 - 0xYE 00 7FFF FFFF: 7A HT1 lo 地址空间  
 0xYE 80 0000 0000 - 0xYE FC FFFF FFFF: 连接到处理器节点 Y 的扩展 PCI MEM 空间  
 0xYE FD 0000 0000 - 0xYE FD FFFF FFFF: 保留  
 0xYE FE 0000 0000 - 0xYE FF FFFF FFFF: 连接到处理器节点 Y 的扩展 PCI 配置空间  
 0xYF 00 0000 0000 - 0xYF FF FFFF FFFF: 保留

以节点 0 地址为例，龙芯 5000 系列处理器+7A 芯片组地址空间的详细分布如表 1-8:

表 1-8 龙芯 5000 系列处理器+7A 芯片组地址空间分布

	起始地址	结束地址	说明
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	内存
地址 1	0x0000_0000_1000_0000	0x0000_0000_17FF_FFFF	32 位模式下 7A 设备寄存器空间
地址 2	0x0000_0000_1800_0000	0x0000_0000_1801_FFFF	32 位模式下 7A LPC 的 IO 空间
地址 3	0x0000_0000_1802_0000	0x0000_0000_19FF_FFFF	32 位模式下 7A PCI 的 IO 空间
地址 4	0x0000_0000_1A00_0000	0x0000_0000_1BFF_FFFF	32 位模式下 7A PCI 的配置空间
地址 5	0x0000_0000_1C00_0000	0x0000_0000_1DFF_FFFF	SPI Memory1
地址 6	0x0000_0000_1FC0_0000	0x0000_0000_1FCF_FFFF	SPI Memory0
地址 7	0x0000_0000_1FE0_01E0	0x0000_0000_1FE0_01E7	UART 0
地址 8	0x0000_0000_1FE0_01E8	0x0000_0000_1FE0_01EF	UART 1
地址 9	0x0000_0000_1FE0_01F0	0x0000_0000_1FE0_01FF	SPI Register
地址 10	0x0000_0000_3000_0000	0x0000_0000_7FFF_FFFF	7A 中 PCI MEM 空间
地址 11	0x0000_0000_8000_0000	0x0000_0000_8FFF_FFFF	保留
地址 12	0x0000_0000_9000_0000	MEM_MAX_ADDR	内存
地址 13	0x0000_0E00_1000_0000	0x0000_0E00_1000_0FFF	7A 中断控制器空间
地址 14	0x0000_0E00_1000_1000	0x0000_0E00_1000_1FFF	7A 中 HPET 寄存器空间
地址 15	0x0000_0E00_1000_2000	0x0000_0E00_1000_2FFF	7A 中 LPC 控制寄存器空间
地址 16	0x0000_0E00_1001_0000	0x0000_0E00_1001_FFFF	7A 中 confbus 空间
地址 17	0x0000_0E00_1008_0000	0x0000_0E00_100F_FFFF	7A 中 misc 设备寄存器空间
地址 18	0x0000_0E00_1200_0000	0x0000_0E00_13FF_FFFF	7A 中 LPC MEM 空间
地址 19	0x0000_0E00_2000_0000	0x0000_0EFC_FFFF_FFFF	7A 中 PCI MEM 空间
地址 20	0x0000_0EFD_FC00_0000	0x0000_0EFD_FDFF_FFFF	HT1 的 I/O 空间
地址 21	0x0000_0EFD_FE00_0000	0x0000_0EFD_FFFF_FFFF	HT1 的总线配置空间



## 2 参考配置

### 2.1 MADT 配置

表 2-1 MADT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'APIC'
Revision	1	8	1
Local Interrupt Controller Address	4	36	0x1fe01400

表 2-2 CORE PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x11
Length	1	1	15
Version	1	2	1
ACPI Processor ID	4	3	处理器核 UID, 与 DSDT 处理器对象中的 _UID 值相同
Physical Processor ID	4	7	CPU 核物理 ID
Flags	4	11	CORE PIC 的标志, 参考表 2-3

表 2-3 CORE PIC 标志

CORE PIC Flags	大小 (比特)	偏移 (比特)	描述
Enabled	1	0	0: CPU 不可用 1: CPU 可用
Reserved	31	1	必须为 0

MADT 表的 LIO PIC 结构参考如下表 2-4:

表 2-4 LIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
---	------------	------------	----

Type	1	0	0x12, LIO PIC 结构
Length	1	1	23
Version	1	2	1
Base Address	8	3	0x1FE01400
Size	2	11	0x80
Cascade vector	2	13	0x0002
Cascade vector mapping	8	15	0x000000000000FFFFFF

MADT 表的 EIO PIC 结构参考如下表 2-5，每个芯片组连接的处理器节点对应一个 EIO PIC 结构：

表 2-5 EIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x14, EIO PIC 结构
Length	1	1	4
Version	1	2	1
Cascade vector	1	3	3+N, N 为 EIO PIC 结构从 0 开始的序号
Node	1	4	连接芯片组的处理器节点 ID
Node map	8	5	EIO 中断路由的处理器节点组, bit0-63 分别表示 0-63 节点

注：EIO PIC 结构中描述的节点表示扩展 IO 中断节点的概念，详见相关处理器手册扩展 IO 中断章节。

MADT 表的 MSI PIC 结构参考如下表 2-6，每个芯片组对应一个相同的 MSI PIC 结构：

表 2-6 MSI PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x15, MSI PIC 结构
Length	1	1	19
Version	1	2	1
Message Address	8	3	0x2FF00000
Start	4	11	0x40
Count	4	15	0xc0

MADT 表的 BIO PIC 结构参考如下表 2-7：

表 2-7 BIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x16, BIO PIC 结构

Length	1	1	17
Version	1	2	1
Base Address	8	3	0xE0010000000
Size	2	11	0x1000
Hardware ID	2	13	0
GSI base	2	15	0x40

对于双桥片的主板（以桥 0 连接到 0 节点，桥 1 连接到 N 节点为例），需要再添加一组 BIO PIC 结构的描述，参考如下表 2-8：

表 2-8 BIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x16, BIO PIC 结构
Length	1	1	17
Version	1	2	1
Base Address	8	3	0xNE0010000000
Size	2	11	0x1000
Hardware ID	2	13	N
GSI base	2	15	0x80

MADT 表的 LPC PIC 结构参考如下表 2-9：

表 2-9 LPC PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x17, LPC PIC 结构
Length	1	1	15
Version	1	2	1
Base Address	8	3	0xE0010002000
Size	2	11	0x1000
Cascade vector	2	13	0x13

## 2.2 FADT 配置

此表为操作系统提供了 Fixed 硬件 ACPI 描述信息，相关约束见表 2-10。

表 2-10 FADT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			

Signature	4	0	'FACP'
Length	4	4	表的长度，包括表头，从偏移量 0 开始。用于记录整个表的大小
FADT Major Version	1	8	3
FIRMWARE_CTRL	4	36	0
DSDT	4	40	0
Reserved	1	44	0
SCI_INT	2	46	0x6f
SMI_CMD	4	48	0
ACPI_ENABLE	1	52	0
ACPI_DISABLE	1	53	0
S4BIOS_REQ	1	54	0
PSTATE_CNT	1	55	0
PM1a_EVT_BLK	4	56	0
PM1b_EVT_BLK	4	60	0
PM1a_CNT_BLK	4	64	0
PM1b_CNT_BLK	4	68	0
PM2_CNT_BLK	4	72	0
PM_TMR_BLK	4	76	0
GPE0_BLK	4	80	0
GPE1_BLK	4	84	0
PM1_EVT_LEN	1	88	8
PM1_CNT_LEN	1	89	4
PM2_CNT_LEN	1	90	0
PM_TMR_LEN	1	91	4
GPE0_BLK_LEN	1	92	8
GPE1_BLK_LEN	1	93	0
GPE1_BASE	1	94	0
CST_CNT	1	95	0
P_LVL2_LAT	2	96	0x65
P_LVL3_LAT	2	98	0x3e9
FLUSH_SIZE	2	100	0
FLUSH_STRIDE	2	102	0
DUTY_OFFSET	1	104	0
DUTY_WIDTH	1	105	0
DAY_ALARM	1	106	0
MON_ALARM	1	107	0
CENTURY	1	108	0
IAPC_BOOT_ARCH	2	109	0

Reserved	1	111	0
Flags	4	112	默认值为 0x425，包含如下标志： WBINVD、PROC_C1、SLP_BUTTON、RESET_REG_SUP
RESET_REG	12	116	见表 2-11
RESET_VALUE	1	128	0x01
ARM_BOOT_ARCH	2	129	0
FADT Minor Version	1	131	0
X_FIRMWARE_CTRL	8	132	FACS 64bit 地址，固件动态生成
X_DSDT	8	140	DSDT 64bit 地址，固件动态生成
X_PM1a_EVT_BLK	12	148	见表 2-12
X_PM1b_EVT_BLK	12	160	此结构中成员全为 0
X_PM1a_CNT_BLK	12	172	见表 2-13
X_PM1b_CNT_BLK	12	184	此结构中成员全为 0
X_PM2_CNT_BLK	12	196	此结构中成员全为 0
X_PM_TMR_BLK	12	208	见表 2-14
X_GPE0_BLK	12	220	见表 2-15
X_GPE1_BLK	12	232	此结构中成员全为 0

注：UEFI 固件动态生成 X\_FIRMWARE\_CTRL 与 X\_DSDT 之前，需将其初始化为 0。

表 2-11 reset 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0
Reg Bit Width	1	1	0x20
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000e00100d0030

表 2-12 PM1a EVT 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x40
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000e00100d000c

表 2-13 PM1a\_CNT 寄存器结构

域	大小	偏移	描述
---	----	----	----

	(字节)	(字节)	
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x20
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000e00100d0014

表 2-14 PM\_TMR 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x20
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000e00100d0018

表 2-15 GPE0 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x40
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000e00100d0028

## 2.3 DSDT 配置

### 2.3.1 PCI 总线枚举

\_SEG (Segment) 方法描述了 PCI 段号。

\_CRS (Current Resource Settings) 方法支持总线范围、IO 地址范围以及内存地址范围声明。

\_OSC (Operating System Capabilities) 方法, 当配置了 \_OSC 时, 仅支持 AER、PME 和 PCI Express Capability Structure control。

(1) PCIE 单主桥配置参考如下:

- \_SEG 为 0
- \_CRS

使用 WordBusNumber()宏声明 PCI 总线号范围, 如表 2-16:

表 2-16 PCI 总线号范围

参数名	值
ResourceUsage	ResourceProducer
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Decode	PosDecode
AddressGranularity	0x0
AddressMinimum	0x0
AddressMaximum	0x00FF
AddressTranslation	0x0000
RangeLength	0x0100
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空

使用 QWordIO()宏声明 PCI I/O 地址范围，如表 2-17：

表 2-17 PCI I/O 地址范围

参数名	值
ResourceUsage	ResourceProducer
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Decode	PosDecode
ISARanges	EntireRange
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000000000000000
AddressMaximum	0x000000000000FFFF
AddressTranslation	0x0000000018000000
RangeLength	0x000000000010000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
TranslationType	值为空
TranslationDensity	值为空

使用 QWordMemory()宏声明 32 位 PCI MEM 地址范围如表 2-18：

表 2-18 32 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer

Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000000030000000
AddressMaximum	0x000000007FFFFFFF
AddressTranslation	0x00000E0000000000
RangeLength	0x0000000050000000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

使用 QWordMemory()宏声明 64 位 PCI MEM 地址范围如表 2-19:

表 2-19 64 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000008000000000
AddressMaximum	0x000000FCFFFFFFF
AddressTranslation	0x00000E0000000000
RangeLength	0x0000007D00000000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

(2) PCIE 双主桥配置参考如下:



主桥一的配置：

- `_SEG` 为 0
- `_CRS`

使用 `WordBusNumber()`宏声明 PCI 总线号范围，如表 2-20：

表 2-20 PCI 总线号范围

参数名	值
<code>ResourceUsage</code>	<code>ResourceProducer</code>
<code>IsMinFixed</code>	<code>MinFixed</code>
<code>IsMaxFixed</code>	<code>MaxFixed</code>
<code>Decode</code>	<code>PosDecode</code>
<code>AddressGranularity</code>	<code>0x0</code>
<code>AddressMinimum</code>	<code>0x0</code>
<code>AddressMaximum</code>	<code>0x00FF</code>
<code>AddressTranslation</code>	<code>0x0000</code>
<code>RangeLength</code>	<code>0x0100</code>
<code>ResourceSourceIndex</code>	值为空
<code>ResourceSource</code>	值为空
<code>DescriptorName</code>	值为空

使用 `QWordIO()`宏声明 PCI I/O 地址范围，如表 2-21：

表 2-21 PCI I/O 地址范围

参数名	值
<code>ResourceUsage</code>	<code>ResourceProducer</code>
<code>IsMinFixed</code>	<code>MinFixed</code>
<code>IsMaxFixed</code>	<code>MaxFixed</code>
<code>Decode</code>	<code>PosDecode</code>
<code>ISARanges</code>	<code>EntireRange</code>
<code>AddressGranularity</code>	<code>0x0000000000000000</code>
<code>AddressMinimum</code>	<code>0x0000000000000000</code>
<code>AddressMaximum</code>	<code>0x000000000000FFFF</code>
<code>AddressTranslation</code>	<code>0x0000000018000000</code>
<code>RangeLength</code>	<code>0x0000000000010000</code>
<code>ResourceSourceIndex</code>	值为空
<code>ResourceSource</code>	值为空
<code>DescriptorName</code>	值为空
<code>TranslationType</code>	值为空
<code>TranslationDensity</code>	值为空

使用 `QWordMemory()`宏声明 32 位 PCI MEM 地址范围，如表 2-22：

表 2-22 32 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000000030000000
AddressMaximum	0x000000007FFFFFFF
AddressTranslation	0x00000E0000000000
RangeLength	0x0000000050000000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

使用 QWordMemory()宏声明 64 位 PCI MEM 地址范围，如表 2-23：

表 2-23 64 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x00000000
AddressMinimum	0x0000008000000000
AddressMaximum	0x000000FCFFFFFFF
AddressTranslation	0x00000E0000000000
RangeLength	0x0000007D00000000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空

TranslationType	值为空
-----------------	-----

主桥二的配置：

- `_SEG` 为 1
- `_CRS`

使用 `WordBusNumber()`宏声明 PCI 总线号范围，如表 2-24：

表 2-24 PCI 总线号范围

参数名	值
ResourceUsage	ResourceProducer
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Decode	PosDecode
AddressGranularity	0x0
AddressMinimum	0x0
AddressMaximum	0x00FF
AddressTranslation	0x0000
RangeLength	0x0100
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空

使用 `QWordIO()`宏声明 PCI I/O 地址范围，如表 2-25：

表 2-25 PCI I/O 地址范围

参数名	值
ResourceUsage	ResourceProducer
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Decode	PosDecode
ISARanges	EntireRange
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000000000000000
AddressMaximum	0x000000000000FFFF
AddressTranslation	0x00005EFDFC000000
RangeLength	0x0000000000010000
ResourceSourceIndex	值为空
ResourceSource	值为空

DescriptorName	值为空
TranslationType	值为空
TranslationDensity	值为空

使用 QWordMemory()宏声明 32 位 PCI MEM 地址范围，如表 2-26:

表 2-26 32 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000000030000000
AddressMaximum	0x000000007FFFFFFF
AddressTranslation	0x0000YE0000000000, Y 代表当前 PCI 主桥连接的处理器节点的节点号
RangeLength	0x0000000050000000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

使用 QWordMemory()宏声明 64 位 PCI MEM 地址范围，如表 2-27:

表 2-27 64 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000008000000000
AddressMaximum	0x000000FCFFFFFFF
AddressTranslation	0x0000YE0000000000, Y 代表当前 PCI 主桥连接的处理器节点的节点号
RangeLength	0x0000007D00000000

ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

## 2.3.2 设备电源管理

7A 芯片组系统支持 USB2.0 和 GMAC 唤醒系统，需要配置相应的\_PRW。

### 2.3.2.1 USB

- \_PRW ( Power Resources for Wake )

表 2-28 package 对象参数表

参数名	值
EventInfo	设备号为 D4:F1 的 USB 控制器：值为 0xa 设备号为 D5:F1 的 USB 控制器：值为 0xd
DeepestSleepState	0x3

### 2.3.2.2 GMAC

- \_PRW ( Power Resources for Wake )

表 2-29 package 对象参数表

参数名	值
EventInfo	设备号为 D3:F0 的 GMAC 控制器：值为 0x5 设备号为 D3:F1 的 GMAC 控制器：值为 0x6
DeepestSleepState	0x3

## 2.3.3 系统休眠唤醒

表 2-30 SLP\_TYP 规定

芯片组	休眠状态	SLP_TYP
7A1000、 7A2000	S0	0
	S3	5
	S4	6
	S5	7

## 2.3.4 串口配置

- `_CRS`

表 2-31 串口参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	处理器串口 0 值为: 0x000000001FE001E0 芯片组串口 0 值为: 0x00000E0010080000
AddressMaximum	处理器串口 0 值为: 0x000000001FE001E7 芯片组串口 0 值为: 0x00000E00100800FF
AddressTranslation	0x0
RangeLength	处理器串口 0 值为: 0x0000000000000008 芯片组串口 0 值为: 0x0000000000000100
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

中断号资源使用 `Interrupt()`宏声明，各个参数的规定如表 2-32：

表 2-32 `Interrupt()`宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Shared
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	处理器串口 0: 26 芯片组串口: 72

- `_DSD`

表 2-33 支持属性表

属性名	值	说明
clock-frequency	处理器串口值为：100000000 芯片组串口值为：50000000	表示实际外接的串口时钟频率，单位为 Hz

### 2.3.5 GPIO 配置

- `_HID`

(1) 7A1000 芯片组集成的普通 GPIO 为 LOON0002，3A5000 处理器集成的 GPIO 为 LOON0007，7A2000 芯片组集成的普通 GPIO 为 LOON000D，7A2000 芯片组集成的 ACPI GPIO 为 LOON000E，3A6000 处理器集成的 GPIO 为 LOON000F。

- `_CRS`

表 2-34 QWordMemory()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	7A 芯片组集成的普通 GPIO 值为：0x00000E00100E0000 7A 芯片组集成的 ACPI GPIO 值为：0x00000E00100D0080 3A 处理器集成的 GPIO 值为：0x000000001FE00500
AddressMaximum	7A 芯片组集成的普通 GPIO 值为：0x00000E00100E0BFF 7A 芯片组集成的 ACPI GPIO 值为：0x00000E00100D009F 3A 处理器集成的 GPIO 值为：0x000000001FE00517
AddressTranslation	0x0
RangeLength	7A 芯片组集成的普通 GPIO 值为：0x0000000000000C00 7A 芯片组集成的 ACPI GPIO 值为：0x0000000000000020 3A 处理器集成的 GPIO 值为：0x0000000000000018
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

每个中断号资源使用一个 `Interrupt()`宏声明，各个参数的规定如表 2-35：

表 2-35 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Shared
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	7A 芯片组集成的普通 GPIO: 123、124、125、126、127 7A 芯片组集成的 ACPI GPIO: 111 3A 处理器 GPIO: 16、17、18、19、20、21、22、23

● \_DSD

表 2-36 支持属性如表

属性名	值	说明
gpio_base	7A 芯片组集成的普通 GPIO 值为: 0x10 3A 处理器集成的 GPIO 值为: 0x50 7A 芯片组集成的 ACPI GPIO 值为: 0x70	GPIO 在内核中的起始编号
ngpios	7A1000 芯片组集成的普通 GPIO 值为: 57 7A2000 芯片组集成的普通 GPIO 值为: 58 7A2000 芯片组集成的 ACPI GPIO 值为: 16 3A 处理器集成的 GPIO 值为: 32	当前注册的 gpio 设备包含的 gpio pin 总数



### 2.3.6 I2C 配置

- `_HID`  
(1) LOON0004。

- `_UID`

各 I2C 控制器与 `_UID` 对应关系规定如表 2-37：

表 2-37 I2C 控制器与 `_UID` 对应表

控制器	<code>_UID</code>
I2C 控制器 0	0
I2C 控制器 1	1
I2C 控制器 2	2
I2C 控制器 3	3
I2C 控制器 4	4
I2C 控制器 5	5

- `_CRS`

表 2-38 `QWordMemory()`宏参数表

参数名	值
<code>ResourceUsage</code>	<code>ResourceConsumer</code>
<code>Decode</code>	<code>PosDecode</code>
<code>IsMinFixed</code>	<code>MinFixed</code>
<code>IsMaxFixed</code>	<code>MaxFixed</code>
<code>Cacheable</code>	<code>NonCacheable</code>
<code>ReadAndWrite</code>	<code>ReadWrite</code>
<code>AddressGranularity</code>	<code>0x0</code>
<code>AddressMinimum</code>	I2C 控制器值为： I2C 0: <code>0x00000E0010090000</code> I2C 1: <code>0x00000E0010090100</code> I2C 2: <code>0x00000E0010090200</code> I2C 3: <code>0x00000E0010090300</code> I2C 4: <code>0x00000E0010090400</code> I2C 5: <code>0x00000E0010090500</code>
<code>AddressMaximum</code>	I2C 控制器值为： I2C 0: <code>0x00000E0010090007</code> I2C 1: <code>0x00000E0010090107</code> I2C 2: <code>0x00000E0010090207</code> I2C 3: <code>0x00000E0010090307</code> I2C 4: <code>0x00000E0010090407</code>

	I2C 5: 0x00000E0010090507
AddressTranslation	0x0
RangeLength	I2C 控制器值为: I2C 0: 0x0000000000000008 I2C 1: 0x0000000000000008 I2C 2: 0x0000000000000008 I2C 3: 0x0000000000000008 I2C 4: 0x0000000000000008 I2C 5: 0x0000000000000008
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

### 2.3.7 RTC 配置

- `_HID`  
LOON0001。
- `_CRS`

表 2-39 QWordMemory()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	0x00000E00100d0100
AddressMaximum	0x00000E00100d01FF
AddressTranslation	0x0
RangeLength	0x0000000000000100
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

中断号资源使用 Interrupt()宏声明，各个参数的规定如表 2-40：

表 2-40 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Exclusive
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	116

### 2.3.8 PWM 配置

- `_HID`  
(1) LOON0006。
- `_CRS`

表 2-41 QWordMemory()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	PWM 0: 0x00000E00100A0000 PWM 1: 0x00000E00100A0100 PWM 2: 0x00000E00100A0200 PWM 3: 0x00000E00100A0300
AddressMaximum	PWM 0: 0x00000E00100A000F PWM 1: 0x00000E00100A010F PWM 2: 0x00000E00100A020F PWM 3: 0x00000E00100A030F
AddressTranslation	0x0
RangeLength	PWM 0: 0x0000000000000010 PWM 1: 0x0000000000000010

	PWM 2: 0x0000000000000010 PWM 3: 0x0000000000000010
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

中断号资源使用 Interrupt()宏声明，各个参数的规定如表 2-42：

表 2-42 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Shared
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	PWM 0: 88 PWM 1: 89 PWM 2: 90 PWM 3: 91

### 2.3.9 热区管理

当配置龙芯 5000 系列处理器温度传感器时，可参考如下部分示例：

```
Scope (\_SB)
{
    Name (IDDR,0x1fe00000) //ChipReg

    OperationRegion (BASE, SystemMemory, IDDR, 0x1a4)
    Field (BASE, AnyAcc, NoLock, Preserve)
    {
        Offset (0x19C), //thsens
        THSE, 32
    }
    .....
}
```

```
Scope (\_TZ)
{
  ThermalZone (THM0)
  {
    Method (_TMP, 0, NotSerialized) // _TMP: Temperature
    {
      Store (\_SB.THSE, Local0)
      Store (CCNT (Local0), Local1)
      Return (C2K (Local1))
    }
    ... ..
  }

  Method (CCNT, 1, NotSerialized) //calculate cpu node temp, (reg & 0xffff) * 731 / 0x4000 - 273;
  {
    Multiply (And (Arg0, 0xFFFF), 0x02DB, Local0)
    Divide (Local0, 0x4000, , Local1)
    Subtract (Local1, 273, Local2)

    Return (Local2)
  }

  Method (C2K, 1, NotSerialized) //Celsius to Kelvin
  {
    Add (Multiply (Arg0, 0x0A), 0x0AAC, Local0)
    If (LLessEqual (Local0, 0x0AAC))
    {
      Store (0x0BB8, Local0)
    }

    If (LGreater (Local0, 0x0FAC))
    {
      Store (0x0BB8, Local0)
    }

    Return (Local0)
  }
}
```

当配置 7A2000 温度传感器时，可以参考如下部分示例：

```
Scope (\_SB)
{
  Device (PCI0)
  {
```

```
Name (BTSA,0xe0010000400) //bridge temp sensor addr
OperationRegion (BASE, SystemMemory, BTSA, 0x18)
Field (BASE, AnyAcc, NoLock, Preserve)
{
    Offset (0x14), //thsens
    THS1, 32
}
.....
}
}

Scope (\_TZ)
{
    ... ..
    Method (CBST, 1, NotSerialized) //calculate bridge sensor temp, (reg & 0xffff0000) >> 24;
    {
        ShiftRight (And (Arg0, 0xFFFF0000), 24, Local0)
        Return (Local0)
    }

    ThermalZone (THM1)
    {
        Name(_TZP, 300) //polling delay

        Method (_TMP, 0, NotSerialized) // _TMP: Temperature
        {
            Store (\_SB.PCI0.THS1, Local0)
            Store (CBST (Local0), Local1)
            Return (C2K (Local1))
        }

        Method (_CRT, 0, NotSerialized) // _CRT: Critical Temperature
        {
            Return (C2K (0x60))
        }
    }
    ... ..
}
```

### 2.3.10 SE 设备

- `_HID`
  - (1) LOON0003。
- `_CRS`

中断号资源使用 `Interrupt()`宏声明，各个参数的规定如表 2-43：

表 2-43 `Interrupt()`宏参数表

参数名	值
<code>ResourceUsage</code>	<code>ResourceConsumer</code>
<code>EdgeLevel</code>	<code>Level</code>
<code>ActiveLevel</code>	<code>ActiveHigh</code>
<code>Shared</code>	<code>Exclusive</code>
<code>ResourceSourceIndex</code>	值为空
<code>ResourceSource</code>	值为空
<code>DescriptorName</code>	值为空
<code>InterruptList</code>	33、36

### 2.3.11 EC 配置

- `_GPE`
  - (1) 对于 7A1000 芯片组，SCI 中断对应的 ACPI LID 管脚位
  - (2) 对于 7A2000 芯片组，SCI 中断对应的 ACPI GPIO 管脚位

### 2.3.12 SPI 配置

当配置 7A 芯片组 SPI 控制器时，可以参考如下部分示例：

```
Device (SPI0)
{
    Name (_ADR, 0x00160000) // _ADR: Address
    Name (_DSD, Package (0x02) // _DSD: Device-Specific Data
    {
        ToUUID ("daffd814-6eba-4d8c-8a91-bc9bbf4aa301") /* Device Properties for _DSD */,
        Package (0x01)
        {
            Package (0x02)
            {
                "clock-frequency",
                100000000
            }
        }
    }
}
```

```

    }
  })
}

```

## 2.4 FACS 配置

表 2-44 FACS 表约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'FACS'
Firmware Waking Vector	4	12	0
Global Lock	4	16	0
Flags	4	20	0
X Firmware Waking Vector	8	24	0
Version	1	32	1

## 2.5 S3 休眠地址

SADR，是系统 S3 休眠时进入固件执行休眠的入口地址，该地址由固件决定，存放固件的 S3 处理程序。

## 2.6 SRAT 配置

表 2-45 SRAT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'SRAT'
Revision	1	8	2

表 2-46 Processor Local APIC/SAPIC Affinity 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0，代表结构为 Processor Local APIC/SAPIC Affinity Structure。
Length	1	1	16
Proximity Domain [7:0]	1	2	处理器节点号的 0-7 位



APIC ID	1	3	处理器 Local APIC ID, 见 MADT 表
Flags	4	4	标志。 0 位: 0 表示该 Processor Local APIC/SAPIC Affinity Structure 不可用; 1 表示该 Processor Local APIC/SAPIC Affinity Structure 可用。 [31:1]位: 必须为 0
Local SAPIC EID	1	8	0
Proximity Domain [31:8]	3	9	处理器节点号的 8-31 位
Clock Domain	4	12	0

表 2-57 Memory Affinity 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	1, 代表结构为 Memory Affinity Structure
Length	1	1	40
Proximity Domain	4	2	内存范围所属的节点号
Reserved	2	6	保留
Base Address Low	4	8	内存范围地址的低 32 位
Base Address High	4	12	内存范围地址的高 32 位
Length Low	4	16	内存范围大小的低 32 位
Length High	4	20	内存范围大小的高 32 位
Reserved	4	24	保留
Flags	4	28	标志, 指示内存区域是否已启用并可以热插拔, 见表 2-48
Reserved	8	32	保留

表 2-48 内存热插拔标志

域	大小 (比特)	偏移 (比特)	描述
Enabled	1	0	0: 该 Memory Affinity Structure 不可用 1: 该 Memory Affinity Structure 可用
Hot Pluggable	1	1	0
NonVolatile	1	2	0
Reserved	29	3	0

## 2.7 MCFG 配置

如为双桥服务器设备（以桥 0 连接到 0 节点，桥 1 连接到 5 节点为例），MCFG 表的配置如表 2-49、2-50、2-51 所示。

表 2-49 MCFG 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'MCFG'
Revision	1	8	1
OEMID	6	10	'LOONGS'
OEM Table ID	8	16	'LOONGSON'
OEM Revision	4	24	1

表 2-50 Memory Mapped Enhanced Configuration Space Base Address Allocation Structure[0]结构

域	大小 (字节)	偏移 (字节)	描述
Base Address	8	0	0x00000efe00000000
PCI Segment Group Number	2	8	0
Start Bus Number	1	10	0x00
End Bus Number	1	11	0xFF
Reserved	4	12	0

表 2-51 Memory Mapped Enhanced Configuration Space Base Address Allocation Structure[1]结构

域	大小 (字节)	偏移 (字节)	描述
Base Address	8	0	0x00005efe00000000
PCI Segment Group Number	2	8	1
Start Bus Number	1	10	0x00
End Bus Number	1	11	0xFF
Reserved	4	12	0

## 2.8 SLIT 配置

表 2-52 SLIT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'SLIT'
Number of System Localities	8	36	节点数。 单路: 0x1; 双路: 0x2; 四路: 0x4; 八路: 0x8; 十六路: 0x10
Entry[Number of System Localities-1][Number of System Localities-1]	1		根据龙芯参考代码实现

## 2.9 SPCR 配置

表 2-53 SPCR 表约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'SPCR'
Interface Type	1	36	0
Reserved	3	37	0
Space ID	1	40	0
Bit Width	1	41	0
Bit Offset	1	42	0
Encoded Access Width	1	43	1
Address	8	44	0x000000001FE001E0
Interrupt Type	1	52	0
PCAT-compatible IRQ	1	53	0
Interrupt	4	54	0
Baud Rate	1	58	0
Parity	1	59	0
Stop Bits	1	60	0
Flow Control	1	61	0

Terminal Type	1	62	0
Language	1	63	0
PCI Device ID	2	64	0xFFFF
PCI Vendor ID	2	66	0xFFFF
PCI Bus	1	68	0
PCI Device	1	69	0
PCI Function	1	70	0
PCI Flags	4	71	0
PCI Segment	1	75	0
Reserved	4	76	0

## 2.10 PPTT 配置

表 2-54 PPTT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'PPTT'
Revision	1	8	3

表 2-55 Processor hierarchy node structure 约束

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0
Length	1	1	14
Reserved	2	2	0
Flags	4	4	处理器结构标志
Parent	4	8	当有父处理器节点结构时，值为 PPTT 表的开始和父处理器结构项的开始之间的差值； 当没有父处理器节点结构时，值为 0
ACPI Processor ID	4	12	MADT 表中的处理器核 UID
Number of private resources	4	16	0

以 3A6000+7A2000 为例，Processor hierarchy node structure 结构约束如下：

[024h 0036 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[025h 0037 001h]	Length : 14
[026h 0038 002h]	Reserved : 0000
[028h 0040 004h]	Flags (decoded below) : 00000011
	Physical package : 1
	ACPI Processor ID valid : 0

	Processor is a thread : 0
	Node is a leaf : 0
	Identical Implementation : 1
[02Ch 0044 004h]	Parent : 00000000
[030h 0048 004h]	ACPI Processor ID : 00000000
[034h 0052 004h]	Private Resource Number : 00000000
[038h 0056 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[039h 0057 001h]	Length : 14
[03Ah 0058 002h]	Reserved : 0000
[03Ch 0060 004h]	Flags (decoded below) : 00000012
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 0
	Node is a leaf : 0
	Identical Implementation : 1
[040h 0064 004h]	Parent : 00000024
[044h 0068 004h]	ACPI Processor ID : 00000000
[048h 0072 004h]	Private Resource Number : 00000000
[04Ch 0076 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[04Dh 0077 001h]	Length : 14
[04Eh 0078 002h]	Reserved : 0000
[050h 0080 004h]	Flags (decoded below) : 0000001E
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 1
	Node is a leaf : 1
	Identical Implementation : 1
[054h 0084 004h]	Parent : 00000038
[058h 0088 004h]	ACPI Processor ID : 00000001
[05Ch 0092 004h]	Private Resource Number : 00000000
[060h 0096 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[061h 0097 001h]	Length : 14
[062h 0098 002h]	Reserved : 0000
[064h 0100 004h]	Flags (decoded below) : 0000001E
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 1
	Node is a leaf : 1
	Identical Implementation : 1
[068h 0104 004h]	Parent : 00000038
[06Ch 0108 004h]	ACPI Processor ID : 00000002

[070h 0112 004h]	Private Resource Number : 00000000
[074h 0116 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[075h 0117 001h]	Length : 14
[076h 0118 002h]	Reserved : 0000
[078h 0120 004h]	Flags (decoded below) : 00000012
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 0
	Node is a leaf : 0
	Identical Implementation : 1
[07Ch 0124 004h]	Parent : 00000024
[080h 0128 004h]	ACPI Processor ID : 00000001
[084h 0132 004h]	Private Resource Number : 00000000
[088h 0136 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[089h 0137 001h]	Length : 14
[08Ah 0138 002h]	Reserved : 0000
[08Ch 0140 004h]	Flags (decoded below) : 0000001E
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 1
	Node is a leaf : 1
	Identical Implementation : 1
[090h 0144 004h]	Parent : 00000074
[094h 0148 004h]	ACPI Processor ID : 00000003
[098h 0152 004h]	Private Resource Number : 00000000
[09Ch 0156 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[09Dh 0157 001h]	Length : 14
[09Eh 0158 002h]	Reserved : 0000
[0A0h 0160 004h]	Flags (decoded below) : 0000001E
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 1
	Node is a leaf : 1
	Identical Implementation : 1
[0A4h 0164 004h]	Parent : 00000074
[0A8h 0168 004h]	ACPI Processor ID : 00000004
[0ACh 0172 004h]	Private Resource Number : 00000000
[0B0h 0176 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[0B1h 0177 001h]	Length : 14
[0B2h 0178 002h]	Reserved : 0000

[0B4h 0180 004h]	Flags (decoded below) : 00000012 Physical package : 0 ACPI Processor ID valid : 1 Processor is a thread : 0 Node is a leaf : 0 Identical Implementation : 1
[0B8h 0184 004h]	Parent : 00000024
[0BCh 0188 004h]	ACPI Processor ID : 00000002
[0C0h 0192 004h]	Private Resource Number : 00000000
[0C4h 0196 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[0C5h 0197 001h]	Length : 14
[0C6h 0198 002h]	Reserved : 0000
[0C8h 0200 004h]	Flags (decoded below) : 0000001E Physical package : 0 ACPI Processor ID valid : 1 Processor is a thread : 1 Node is a leaf : 1 Identical Implementation : 1
[0CCh 0204 004h]	Parent : 000000B0
[0D0h 0208 004h]	ACPI Processor ID : 00000005
[0D4h 0212 004h]	Private Resource Number : 00000000
[0D8h 0216 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[0D9h 0217 001h]	Length : 14
[0DAh 0218 002h]	Reserved : 0000
[0DCh 0220 004h]	Flags (decoded below) : 0000001E Physical package : 0 ACPI Processor ID valid : 1 Processor is a thread : 1 Node is a leaf : 1 Identical Implementation : 1
[0E0h 0224 004h]	Parent : 000000B0
[0E4h 0228 004h]	ACPI Processor ID : 00000006
[0E8h 0232 004h]	Private Resource Number : 00000000
[0ECh 0236 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[0EDh 0237 001h]	Length : 14
[0EEh 0238 002h]	Reserved : 0000
[0F0h 0240 004h]	Flags (decoded below) : 00000012 Physical package : 0 ACPI Processor ID valid : 1 Processor is a thread : 0 Node is a leaf : 0

	Identical Implementation : 1
[0F4h 0244 004h]	Parent : 00000024
[0F8h 0248 004h]	ACPI Processor ID : 00000003
[0FCh 0252 004h]	Private Resource Number : 00000000
[100h 0256 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[101h 0257 001h]	Length : 14
[102h 0258 002h]	Reserved : 0000
[104h 0260 004h]	Flags (decoded below) : 0000001E
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 1
	Node is a leaf : 1
	Identical Implementation : 1
[108h 0264 004h]	Parent : 000000EC
[10Ch 0268 004h]	ACPI Processor ID : 00000007
[110h 0272 004h]	Private Resource Number : 00000000
[114h 0276 001h]	Subtable Type : 00 [Processor Hierarchy Node]
[115h 0277 001h]	Length : 14
[116h 0278 002h]	Reserved : 0000
[118h 0280 004h]	Flags (decoded below) : 0000001E
	Physical package : 0
	ACPI Processor ID valid : 1
	Processor is a thread : 1
	Node is a leaf : 1
	Identical Implementation : 1
[11Ch 0284 004h]	Parent : 000000EC
[120h 0288 004h]	ACPI Processor ID : 00000008
[124h 0292 004h]	Private Resource Number : 00000000



# 龙芯 2K2000 固件开发规范

**V1.0**

## 版本信息

文档信息	文档名	龙芯 2K2000 固件开发规范
	版本号	V1.0
	创建人	系统研发部
版本历史		
序号	版本号	更新内容
1	V1.0	发布文档初始版本 V1.0 版。

目录

前 言.....	1
1 设计约束.....	2
1.1 Audio.....	2
1.1.1 HDA.....	2
1.1.2 I2S.....	2
1.2 VBIOS.....	2
1.3 中断.....	2
1.4 地址空间.....	5
2 参考配置.....	6
2.1 MADT 配置.....	6
2.2 FADT 配置.....	8
2.3 DSDT 配置.....	11
2.3.1 PCI 总线枚举.....	11
2.3.2 设备电源管理.....	13
2.3.3 系统休眠唤醒.....	14
2.3.4 串口配置.....	14
2.3.5 GPIO 配置.....	15
2.3.6 I2C 配置.....	17
2.3.7 RTC 配置.....	18
2.3.8 PWM 配置.....	19
2.3.9 热区管理.....	20

2.3.10 SE 设备.....	21
2.3.11 EC 配置.....	22
2.4 FACS 配置.....	22
2.5 S3 休眠地址.....	22
2.6 SRAT 配置.....	22
2.7 MCFG 配置.....	24
2.9 SPCR 配置.....	24

## 前 言

本规范面向固件软件开发人员，在《龙芯 CPU 通用 PC/服务器系统固件与内核接口规范 V4.1》基础上针对 2K2000 的固件开发约束进行补充说明。

# 1 设计约束

## 1.1 Audio

### 1.1.1 HDA

固件通过 Verb Table 机制实现不同板卡中对声卡的差异化设计。固件开发人员需要根据具体主板的声卡设计正确填写 Verb Table。具体参考龙芯代码中关于 Realtek 的 Verb Table 配置实现。

固件需要根据芯片手册配置 hda\_i2s\_sel 引脚为 HDA 工作模式（芯片配置寄存器偏移 0x0440 寄存器的 12:11 为 x1b，配置寄存器基址需要固件下进行配置，参考代码配置寄存器基址为：0x10010000）。

### 1.1.2 I2S

固件需要根据芯片手册配置 hda\_i2s\_sel 引脚为 I2S 工作模式（芯片配置寄存器偏移 0x0440 寄存器的 12:11 为 10b，配置寄存器基址需要固件下进行配置，参考代码配置寄存器基址为：0x10010000）。

## 1.2 VBIOS

2K2000 内部包含图形处理器（以下简称 GPU），位于 D6:F0；内核下 GPU 需要 Video BIOS（以下简称 VBIOS）的支撑才可以工作正常。

固件对 VBIOS 处理方式：将固件内默认 VBIOS 全部读取到内置显卡 BAR2 空间的最后 1MB。

## 1.3 中断

龙芯 2K2000 方案，南北桥中断控制器管理的中断可通过中断线路至 Node 节点，MSI 中断通过 MSI 消息路由至 Node 节点，以上中断最终路由至处理器核。见图 1-1 所示。

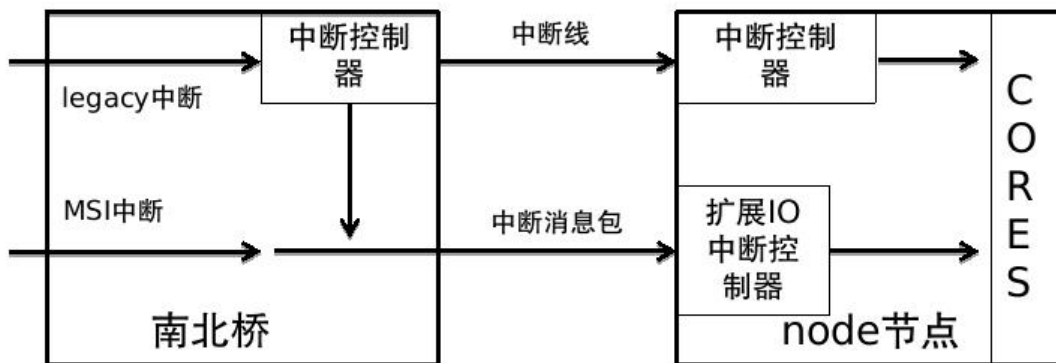


图 1-1 龙芯 2K2000 处理器中断路由示意图

2K2000 系统依据 ACPI 规范使用 GSI (global system interrupts, 全局系统中断) 为中断源分配虚拟化的中断号, 一个 GSI 类型的虚拟中断号唯一标识了其所属的中断域, 内核按照不同的中断域实现中断管理。各个中断域对应的 GSI 中断号范围约定如下:

- 0-15: LPC/ISA 中断域, 用于系统外扩 LPC 设备中断源, 如表 1-1 所示, 具体分配情况参考 LPC 设备说明。

表 1-1 LPC/ISA 中断域 GSI 分配

GSI 中断号	中断源	说明
0	HPET	HPET 高精度定时器
1	I8042	键盘
2	级联	
3		
4		
5		
6		
7		
8	RTC	实时时钟
9		
10		
11		
12	I8042	鼠标
13		
14	ide0	硬盘
15	ide1	硬盘

- 16-47: NODE I/O 中断控制器中断域, 其 GSI 中断号与中断源向量号的映射关系:  
GSI 中断号 = 中断源向量号+16  
中断源向量号为处理器手册中某个中断源在 Node 中断控制寄存器中的索引, 具体参考龙芯 2K2000 处理器手册。
- 48-63: 保留。
- 64-127: 南北桥 I/O 中断控制器中断域。其 GSI 中断号与中断源向量号的映射关系:  
GSI 中断号 = 中断源向量号 + GSI base  
中断源向量号为处理器手册中某个中断源在南北桥中断控制寄存器中的索引, 具体参考龙芯 2K2000 用户手册;  
GSI base 为 MADT 表中当前南北桥 BIO PIC 结构的 GSI base 成员。

其中南北桥集成或外接的 PCI 兼容设备分配的中断号如表 1-2:

表 1-2 南北桥 I/O 中断控制器中断域 GSI 分配

中断号	中断源	中断号	中断源
64	eMMC_DMA	96	pcie_f0_0
65	SDIO_DMA	97	pcie_f0_1
66	rio1	98	pcie_f0_2

67	rio0	99	pcie_f0_3
68	NB_DMA	100	pcie_f1_0
69	MIPHY	101	pcie_f1_1
70	gmac1_phy	102	PWM4/SE_INT6
71	gmac0_phy	103	PWM5
72	UART/SE_INT5	104	PCle2
73	I2C/SE_INT4	105	AES/SE_INT2
74	vpu	106	DES/SE_INT1
75	can	107	RSA/SE_INT0
76	gmac0_sbd	108	toy[0]
77	gmac0_pmt	109	toy[1]
78	gmac1_sbd	110	toy[2]
79	gmac1_pmt	111	acpi
80	sata	112	USB2
81	GMAC2_SBD	113	OTG
82	GMAC2_PMT	114	SDIO
83	lpc	115	eMMC
84	hpet[1]	116	rtc[0]
85	hpet[2]	117	rtc[1]
86	usb3	118	rtc[2]
87	hda1	119	hpet[0]
88	pwm[0]	120	i2s_dma[0]
89	pwm[1]	121	i2s_dma[1]
90	pwm[2]	122	i2s/hda
91	pwm[3]	123	gpio_hi
92	dc	124	gpio[0]/gpio[13]
93	gpu	125	gpio[1]/gpio[14]
94	gmem	126	gpio[2]/gpio[15]
95	thsens	127	gpio[3]/gpio[50]

注意：

(1) 若使用 EC 时需要使用 SCI 中断，请参考 2K2000 硬件规范，此中断需要硬件连接到 2K2000 电源管理模块的 ACPI GPIO 管脚，信号约束参照硬件设计规范，对应 GSI 中断源为 111 号。



## 1.4 地址空间

龙芯 2K2000 地址空间的分布如表 1-3:

表 1-3 龙芯 2K2000 地址空间分布

	起始地址	结束地址	说明
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	内存
地址 1	0x0000_0000_1000_0000	0x0000_0000_17FF_FFFF	用于映射内部所使用的设备空间(BAR)
地址 2	0x0000_0000_1800_0000	0x0000_0000_19FF_FFFF	用于映射 PCIE 控制器对外的 IO 空间
地址 3	0x0000_0000_1A00_0000	0x0000_0000_1AFF_FFFF	Type0 配置空间
地址 4	0x0000_0000_1B00_0000	0x0000_0000_1BFF_FFFF	Type1 配置空间
地址 5	0x0000_0000_1C00_0000	0x0000_0000_1CFF_FFFF	启动空间, 可映射至 SPI 和 LIO 上
地址 6	0x0000_0000_1D00_0000	0x0000_0000_1DFF_FFFF	Local IO Memory
地址 7	0x0000_0000_1FC0_0000	0x0000_0000_1FCF_FFFF	可映射至 SPI、NAND、SDIO 和 LIO 上
地址 8	0x0000_0000_1FE0_0000	0x0000_0000_1FEF_FFFF	芯片配置寄存器空间
地址 9	0x0000_0000_1FFF_0000	0x0000_0000_1FFF_FFFF	SPI 配置空间
地址 10	0x0000_0000_4000_0000	0x0000_0000_7FFF_FFFF	PCIE MEM 空间
地址 11	0x0000_0000_8000_0000	0x0000_0020_7FFF_FFFF	内存
地址 12	0x0000_0040_0000_0000	0x0000_007F_FFFF_FFFF	PCIE MEM 空间
地址 13	0x0000_00FD_FC00_0000	0x0000_00FD_FFFF_FFFF	PCIE I/O 空间
地址 14	0x0000_00FE_0000_0000	0x0000_00FE_0FFF_FFFF	Type0 配置空间
地址 15	0x0000_00FE_1000_0000	0x0000_00FE_1FFF_FFFF	Type1 配置空间
地址 16	0x0000_00FE_2000_0000	0x0000_00FE_2FFF_FFFF	Type0 配置空间
地址 17	0x0000_00FE_3000_0000	0x0000_00FE_3FFF_FFFF	Type1 配置空间

## 2 参考配置

### 2.1 MADT 配置

表 2-1 MADT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'APIC'
Revision	1	8	1
Local Interrupt Controller Address	4	36	0x1fe01400

表 2-2 CORE PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x11
Length	1	1	15
Version	1	2	1
ACPI Processor ID	4	3	处理器核 UID, 与 DSDT 处理器对象中的 _UID 值相同
Physical Processor ID	4	7	CPU 核物理 ID
Flags	4	11	CORE PIC 的标志, 参考表 2-3

表 2-3 CORE PIC 标志

CORE PIC Flags	大小 (比特)	偏移 (比特)	描述
Enabled	1	0	0: CPU 不可用 1: CPU 可用
Reserved	31	1	必须为 0

MADT 表的 LIO PIC 结构参考如下表 2-4:

表 2-4 LIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述

Type	1	0	0x12, LIO PIC 结构
Length	1	1	23
Version	1	2	1
Base Address	8	3	0x1FE01400
Size	2	11	0x80
Cascade vector	2	13	0x0002
Cascade vector mapping	8	15	0x000000000000FFFF

MADT 表的 EIO PIC 结构参考如下表 2-5:

表 2-5 EIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x14, EIO PIC 结构
Length	1	1	4
Version	1	2	1
Cascade vector	1	3	3
Node	1	4	0
Node map	8	5	1

MADT 表的 MSI PIC 结构参考如下表 2-6:

表 2-6 MSI PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x15, MSI PIC 结构
Length	1	1	19
Version	1	2	1
Message Address	8	3	0x1FE01140
Start	4	11	0x40
Count	4	15	0xc0

MADT 表的 BIO PIC 结构参考如下表 2-7:

表 2-7 BIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x16, BIO PIC 结构
Length	1	1	17

Version	1	2	1
Base Address	8	3	0x10000000
Size	2	11	0x1000
Hardware ID	2	13	0
GSI base	2	15	0x40

MADT 表的 LPC PIC 结构参考如下表 2-8:

表 2-8 LPC PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0x17, LPC PIC 结构
Length	1	1	15
Version	1	2	1
Base Address	8	3	0x10002000
Size	2	11	0x1000
Cascade vector	2	13	0x13

## 2.2 FADT 配置

此表为操作系统提供了 Fixed 硬件 ACPI 描述信息, 相关约束见表 2-9。

表 2-9 FADT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'FACP'
Length	4	4	表的长度, 包括表头, 从偏移量 0 开始。用于记录整个表的大小
FADT Major Version	1	8	3
FIRMWARE_CTRL	4	36	0
DSDT	4	40	0
Reserved	1	44	0
SCI_INT	2	46	0x6f
SMI_CMD	4	48	0
ACPI_ENABLE	1	52	0
ACPI_DISABLE	1	53	0
S4BIOS_REQ	1	54	0
PSTATE_CNT	1	55	0
PM1a_EVT_BLK	4	56	0

PM1b_EVT_BLK	4	60	0
PM1a_CNT_BLK	4	64	0
PM1b_CNT_BLK	4	68	0
PM2_CNT_BLK	4	72	0
PM_TMR_BLK	4	76	0
GPE0_BLK	4	80	0
GPE1_BLK	4	84	0
PM1_EVT_LEN	1	88	8
PM1_CNT_LEN	1	89	4
PM2_CNT_LEN	1	90	0
PM_TMR_LEN	1	91	4
GPE0_BLK_LEN	1	92	8
GPE1_BLK_LEN	1	93	0
GPE1_BASE	1	94	0
CST_CNT	1	95	0
P_LVL2_LAT	2	96	0x65
P_LVL3_LAT	2	98	0x3e9
FLUSH_SIZE	2	100	0
FLUSH_STRIDE	2	102	0
DUTY_OFFSET	1	104	0
DUTY_WIDTH	1	105	0
DAY_ALARM	1	106	0
MON_ALARM	1	107	0
CENTURY	1	108	0
IAPC_BOOT_ARCH	2	109	0
Reserved	1	111	0
Flags	4	112	默认值为 0x425，包含如下标志： WBINVD、PROC_C1、SLP_BUTTON、RESET_REG_SUP
RESET_REG	12	116	见表 2-10
RESET_VALUE	1	128	0x01
ARM_BOOT_ARCH	2	129	0
FADT Minor Version	1	131	0
X_FIRMWARE_CTRL	8	132	FACS 64bit 地址，固件动态生成
X_DSDT	8	140	DSDT 64bit 地址，固件动态生成
X_PM1a_EVT_BLK	12	148	见表 2-11
X_PM1b_EVT_BLK	12	160	此结构中成员全为 0
X_PM1a_CNT_BLK	12	172	见表 2-12
X_PM1b_CNT_BLK	12	184	此结构中成员全为 0

X_PM2_CNT_BLK	12	196	此结构中成员全为 0
X_PM_TMR_BLK	12	208	见表 2-13
X_GPE0_BLK	12	220	见表 2-14
X_GPE1_BLK	12	232	此结构中成员全为 0

注：UEFI 固件动态生成 X\_FIRMWARE\_CTRL 与 X\_DSDT 之前，需将其初始化为 0。

表 2-10 reset 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0
Reg Bit Width	1	1	0x20
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000000100d0030

表 2-11 PM1a EVT 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x40
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000000100d000c

表 2-12 PM1a\_CNT 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x20
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000000100d0014

表 2-13 PM\_TMR 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x20
Reg Bit Offset	1	2	0

Reserved	1	3	0
Address	8	4	0x00000000100d0018

表 2-14 GPE0 寄存器结构

域	大小 (字节)	偏移 (字节)	描述
Address Space Id	1	0	0,
Reg Bit Width	1	1	0x40
Reg Bit Offset	1	2	0
Reserved	1	3	0
Address	8	4	0x00000000100d0028

## 2.3 DSDT 配置

### 2.3.1 PCI 总线枚举

\_SEG (Segment) 方法描述了 PCI 段号。

\_CRS (Current Resource Settings) 方法支持总线范围、IO 地址范围以及内存地址范围声明。

\_OSC (Operating System Capabilities) 方法，当配置了 \_OSC 时，仅支持 AER、PME 和 PCI Express Capability Structure control。

(1) 配置参考如下：

- \_SEG 为 0
- \_CRS

使用 WordBusNumber()宏声明 PCI 总线号范围，如表 2-15：

表 2-15 PCI 总线号范围

参数名	值
ResourceUsage	ResourceProducer
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Decode	PosDecode
AddressGranularity	0x0
AddressMinimum	0x0
AddressMaximum	0x00FF
AddressTranslation	0x0000
RangeLength	0x0100
ResourceSourceIndex	值为空

ResourceSource	值为空
DescriptorName	值为空

使用 QWordIO()宏声明 PCI I/O 地址范围，如表 2-16：

表 2-16 PCI I/O 地址范围

参数名	值
ResourceUsage	ResourceProducer
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Decode	PosDecode
ISARanges	EntireRange
AddressGranularity	0x00000000
AddressMinimum	0x0000000000004000
AddressMaximum	0x0000000000009FFF
AddressTranslation	0x000000FDFC000000
RangeLength	0x0000000000006000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
TranslationType	值为空
TranslationDensity	值为空

使用 DWordMemory()宏声明 32 位 PCI MEM 地址范围如表 2-17：

表 2-17 32 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x00000000
AddressMinimum	0x40000000
AddressMaximum	0x7FFFFFFF
AddressTranslation	0x00000000
RangeLength	0x40000000
ResourceSourceIndex	值为空



ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

使用 QWordMemory()宏声明 64 位 PCI MEM 地址范围如表 2-18:

表 2-18 64 位 PCI MEM 地址范围

参数名	值
ResourceUsage	ResourceProducer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	Cacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0000000000000000
AddressMinimum	0x0000004000000000
AddressMaximum	0x0000007FFFFFFF
AddressTranslation	0x0000000000000000
RangeLength	0x0000004000000000
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	值为空
TranslationType	值为空

## 2.3.2 设备电源管理

2K2000 系统支持 USB2.0/USB3.0 唤醒系统，需要配置相应的\_PRW。

### 2.3.2.1 USB

- \_PRW (Power Resources for Wake)

表 2-19 package 对象参数表

参数名	值
EventInfo	设备号为 D4:F0 的 USB 控制器：值为 0xa 设备号为 D25:F0 的 USB 控制器：值为 0xb

DeepestSleepState	0x3
-------------------	-----

### 2.3.3 系统休眠唤醒

表 2-20 SLP\_TYP 规定

处理器	休眠状态	SLP_TYP
2K2000	S0	0
	S3	5
	S4	6
	S5	7

### 2.3.4 串口配置

- \_CRS

表 2-21 串口参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	Node 节点串口 0 值为: 0x000000001FE001E0 南北桥串口 0 值为: 0x0000000010080000
AddressMaximum	Node 节点串口 0 值为: 0x000000001FE001E7 南北桥串口 0 值为: 0x00000000100800FF
AddressTranslation	0x0
RangeLength	Node 节点串口 0 值为: 0x0000000000000008 南北桥串口 0 值为: 0x0000000000000100
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

中断号资源使用 Interrupt()宏声明，各个参数的规定如表 2-22：

表 2-22 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Shared
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	Node 节点串口 0 值为: 26 南北桥串口值为: 72

● \_DSD

表 2-23 支持属性表

属性名	值	说明
clock-frequency	Node 节点串口值为: 100000000 南北桥串口值为: 50000000	表示实际外接的串口时钟频率, 单位为 Hz

### 2.3.5 GPIO 配置

● \_HID

(1) 2K2000 处理器集成的 GPIO 为 LOON000A, 2K2000 南北桥集成的 GPIO 为 LOON000B, 2K2000 南北桥集成的 ACPI GPIO 为 LOON000C。

● \_CRS

表 2-24 QWordMemory()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	南北桥 GPIO 值为: 0x00000000100E0000 南北桥集成的 ACPI GPIO 值为: 0x00000000100D0080 Node 节点 GPIO 值为: 0x000000001FE00500
AddressMaximum	南北桥 GPIO 值为: 0x00000000100E0BFF

	南北桥集成的 ACPI GPIO 值为: 0x00000000100D009F Node 节点 GPIO 值为: 0x000000001FE00517
AddressTranslation	0x0
RangeLength	南北桥 GPIO 值为: 0x00000000000000C00 南北桥集成的 ACPI GPIO 值为: 0x00000000000000020 Node 节点 GPIO 值为: 0x00000000000000018
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

每个中断号资源一个使用 Interrupt()宏声明, 各个参数的规定如表 2-25:

表 2-25 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Shared
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	南北桥 GPIO: 123、124、125、126、127 南北桥集成的 ACPI GPIO: 111 Node 节点 GPIO: 16、17、18、19、20、21、22、23

● \_DSD

表 2-26 支持属性如表

属性名	值	说明
gpio_base	南北桥 GPIO 值为: 0x10 Node 节点 GPIO 值为: 0x50 南北桥集成的 ACPI GPIO: 0x70	GPIO 在内核中的起始编号
ngpios	南北桥 GPIO 值为: 64 Node 节点 GPIO 值为: 32 南北桥集成的 ACPI GPIO: 16	当前注册的 gpio 设备包含的 gpio pin 总数

### 2.3.6 I2C 配置

- `_HID`

(1) LOON0004。

- `_UID`

各 I2C 控制器与 `_UID` 对应关系规定如表 2-27：

表 2-27 I2C 控制器与 `_UID` 对应表

控制器	<code>_UID</code>
I2C 控制器 0	0
I2C 控制器 1	1

- `_CRS`

表 2-28 `QWordMemory()`宏参数表

参数名	值
<code>ResourceUsage</code>	<code>ResourceConsumer</code>
<code>Decode</code>	<code>PosDecode</code>
<code>IsMinFixed</code>	<code>MinFixed</code>
<code>IsMaxFixed</code>	<code>MaxFixed</code>
<code>Cacheable</code>	<code>NonCacheable</code>
<code>ReadAndWrite</code>	<code>ReadWrite</code>
<code>AddressGranularity</code>	<code>0x0</code>
<code>AddressMinimum</code>	I2C 控制器值为： I2C 0: <code>0x0000000010090000</code> I2C 1: <code>0x0000000010090100</code>
<code>AddressMaximum</code>	I2C 控制器值为： I2C 0: <code>0x0000000010090007</code> I2C 1: <code>0x0000000010090107</code>
<code>AddressTranslation</code>	<code>0x0</code>
<code>RangeLength</code>	I2C 控制器值为： I2C 0: <code>0x0000000000000008</code> I2C 1: <code>0x0000000000000008</code>
<code>ResourceSourceIndex</code>	值为空
<code>ResourceSource</code>	值为空

DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

### 2.3.7 RTC 配置

- `_HID`  
LOON0001。
- `_CRS`

表 2-29 QWordMemory()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	0x00000000100d0100
AddressMaximum	0x00000000100d01FF
AddressTranslation	0x0
RangeLength	0x0000000000000100
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

中断号资源使用 `Interrupt()`宏声明，各个参数的规定如表 2-30：

表 2-30 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Exclusive
ResourceSourceIndex	值为空
ResourceSource	值为空

DescriptorName	值为空
InterruptList	116

### 2.3.8 PWM 配置

- `_HID`
- (1) LOON0006。
- `_CRS`

表 2-31 QWordMemory()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
Decode	PosDecode
IsMinFixed	MinFixed
IsMaxFixed	MaxFixed
Cacheable	NonCacheable
ReadAndWrite	ReadWrite
AddressGranularity	0x0
AddressMinimum	PWM 0: 0x00000000100A0000 PWM 1: 0x00000000100A0100 PWM 2: 0x00000000100A0200 PWM 3: 0x00000000100A0300 PWM 4: 0x00000000100A0400 PWM 5: 0x00000000100A0500
AddressMaximum	PWM 0: 0x00000000100A000F PWM 1: 0x00000000100A010F PWM 2: 0x00000000100A020F PWM 3: 0x00000000100A030F PWM 4: 0x00000000100A040F PWM 5: 0x00000000100A050F
AddressTranslation	0x0
RangeLength	PWM 0: 0x0000000000000010 PWM 1: 0x0000000000000010 PWM 2: 0x0000000000000010 PWM 3: 0x0000000000000010 PWM 4: 0x0000000000000010 PWM 5: 0x0000000000000010
ResourceSourceIndex	值为空

ResourceSource	值为空
DescriptorName	值为空
MemoryRangeType	AddressRangeMemory
TranslationType	TypeStatic

中断号资源使用 Interrupt()宏声明，各个参数的规定如表 2-32：

表 2-32 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level
ActiveLevel	ActiveHigh
Shared	Shared
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	PWM 0: 88 PWM 1: 89 PWM 2: 90 PWM 3: 91 PWM 4: 102 PWM 5: 103

### 2.3.9 热区管理

当配置龙芯 2K2000 处理器温度传感器时，可以参考如下部分示例：

```
Scope (\_SB)
{
    Name (IDDR,0x1fe00000) //ChipReg

    OperationRegion (BASE, SystemMemory, IDDR, 0x438)
    Field (BASE, AnyAcc, NoLock, Preserve)
    {
        Offset (0x428), //thsens
        THSE, 32
    }
    .....
}

Scope (\_TZ)
{
```



```

ThermalZone (THM0)
{
  Method (_TMP, 0, NotSerialized) // _TMP: Temperature
  {
    Store (\_SB.THSE, Local0)
    Return (C2K (Local0))
  }
  Method (_CRT, 0, NotSerialized) // _CRT: Critical Temperature
  {
    Return (C2K (0x60))
  }
  ... ..
}

Method (C2K, 1, NotSerialized) //Celsius to Kelvin
{
  Add (Multiply (Arg0, 0x0A), 0x0AAC, Local0)
  If (LLessEqual (Local0, 0x0AAC))
  {
    Store (0x0BB8, Local0)
  }

  If (LGreater (Local0, 0x0FAC))
  {
    Store (0x0BB8, Local0)
  }

  Return (Local0)
}
}

```

### 2.3.10 SE 设备

- **\_HID**  
(1) LOON0003。
- **\_CRS**

中断号资源使用 Interrupt()宏声明，各个参数的规定如表 2-33：

表 2-33 Interrupt()宏参数表

参数名	值
ResourceUsage	ResourceConsumer
EdgeLevel	Level

ActiveLevel	ActiveHigh
Shared	Exclusive
ResourceSourceIndex	值为空
ResourceSource	值为空
DescriptorName	值为空
InterruptList	107、106、105、95、73、72、102、92

### 2.3.11 EC 配置

- `_GPE`

(1) SCI 中断对应的 ACPI GPIO 管脚位

## 2.4 FACS 配置

表 2-34 FACS 表约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'FACS'
Firmware Waking Vector	4	12	0
Global Lock	4	16	0
Flags	4	20	0
X Firmware Waking Vector	8	24	0
Version	1	32	1

## 2.5 S3 休眠地址

SADR，是系统 S3 休眠时进入固件执行休眠的入口地址，该地址由固件决定，存放固件的 S3 处理程序。

## 2.6 SRAT 配置

表 2-35 SRAT 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'SRAT'
Revision	1	8	2

表 2-36 Processor Local APIC/SAPIC Affinity 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	0, 代表结构为 Processor Local APIC/SAPIC Affinity Structure。
Length	1	1	16
Proximity Domain [7:0]	1	2	0, 代表处理器节点号的 0-7 位
APIC ID	1	3	处理器 Local APIC ID, 见 MADT 表
Flags	4	4	标志。 0 位: 0 表示该 Processor Local APIC/SAPIC Affinity Structure 不可用; 1 表示该 Processor Local APIC/SAPIC Affinity Structure 可用。 [31:1]位: 必须为 0
Local SAPIC EID	1	8	0
Proximity Domain [31:8]	3	9	0, 代表处理器节点号的 8-31 位
Clock Domain	4	12	0

表 2-37 Memory Affinity 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	1, 代表结构为 Memory Affinity Structure
Length	1	1	40
Proximity Domain	4	2	0
Reserved	2	6	保留
Base Address Low	4	8	内存范围地址的低 32 位
Base Address High	4	12	内存范围地址的高 32 位
Length Low	4	16	内存范围大小的低 32 位
Length High	4	20	内存范围大小的高 32 位
Reserved	4	24	保留
Flags	4	28	标志, 指示内存区域是否已启用并可以热插拔, 见表 2-38
Reserved	8	32	保留

表 2-38 内存热插拔标志

域	大小 (比特)	偏移 (比特)	描述
Enabled	1	0	0: 该 Memory Affinity Structure 不可用 1: 该 Memory Affinity Structure 可用

Hot Pluggable	1	1	0
NonVolatile	1	2	0
Reserved	29	3	0

## 2.7 MCFG 配置

MCFG 表的配置如表 2-39、2-40 所示。

表 2-39 MCFG 表约束

域	大小 (字节)	偏移 (字节)	描述
Header			
Signature	4	0	'MCFG'
Revision	1	8	1
OEMID	6	10	'LOONGS'
OEM Table ID	8	16	'LOONGSON'
OEM Revision	4	24	1

表 2-40 Memory Mapped Enhanced Configuration Space Base Address Allocation Structure[0]结构

域	大小 (字节)	偏移 (字节)	描述
Base Address	8	0	0x000000fe00000000
PCI Segment Group Number	2	8	0
Start Bus Number	1	10	0x00
End Bus Number	1	11	0xFF
Reserved	4	12	0

## 2.9 SPCR 配置

表 2-41 SPCR 表约束

域	大小 (字节)	偏移 (字节)	描述
Signature	4	0	'SPCR'
Interface Type	1	36	0
Reserved	3	37	0
Space ID	1	40	0

Bit Width	1	41	0
Bit Offset	1	42	0
Encoded Access Width	1	43	1
Address	8	44	0x000000001FE001E0
Interrupt Type	1	52	0
PCAT-compatible IRQ	1	53	0
Interrupt	4	54	0
Baud Rate	1	58	0
Parity	1	59	0
Stop Bits	1	60	0
Flow Control	1	61	0
Terminal Type	1	62	0
Language	1	63	0
PCI Device ID	2	64	0xFFFF
PCI Vendor ID	2	66	0xFFFF
PCI Bus	1	68	0
PCI Device	1	69	0
PCI Function	1	70	0
PCI Flags	4	71	0
PCI Segment	1	75	0
Reserved	4	76	0

# 龙芯 7A1000 内置显卡软硬件设计规范

## V1.0

## 版本信息

文档信息		文档名	龙芯 7A1000 内置显卡软硬件设计规范
		版本号	V1.0
		创建人	系统研发部
版本历史			
序号	版本号	更新内容	
1	V0.1	发布文档初始版本 V0.1 版。	
2	V0.2	调整编码器连接方案； 增加关于热插拔方案的说明。	
3	V1.0	1.更改显示设备名称为“内置显卡”； 2.本规范适用于龙芯 7A1000 桥片，更改文档名称； 3.第 4 节改为“显示类型和模式”，并完善信号转换、多分辨率支持情况的描述； 4.完善热插拔的轮询和中断相关描述； 5.删除 4.1 节单显示器方案相关描述； 6.更正并完善 DDC 和 EDID 获取相关描述； 7.更新第 5 节 vbios 存储约定的说明； 8.删除第六节，编码器芯片的选型。	

目 录

1 范围 .....	1
2 术语与定义 .....	1
3 架构关系 .....	1
4 显示类型和模式 .....	2
4.1 多屏显示方案 .....	3
4.2 DDC 通道硬件连接方案 .....	4
4.3 获取 EDID .....	4
4.4 热插拔探测 .....	5
4.5 背光控制 .....	5
5 VBIOS 存储约定 .....	5



## 前 言

本规范涉及到龙芯 7A1000 桥片内置显卡，主要介绍龙芯板卡内置显卡固件与硬件之间，固件和内核接口之间的功能划分及相关硬件布线约定。

## 1 范围

本规范规定龙芯 7A1000 内置显卡的硬件设计要求，VBIOS 固件的存放位置等。本规范适用于龙芯 7A1000 桥片主板。建议其它系统厂商遵循此规范开发相关产品。

## 2 术语与定义

本规范所用术语定义如下：

**固件**：Firmware，写入 ROM、EPROM 等非易失存储器中的程序，负责控制和协调集成电路。

**BIOS**：基本输入输出系统，Basic Input Output System，一组固化到主板上一个 ROM 芯片上的程序，它保存着计算机基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。BIOS 与硬件系统集成在一起，也被称为固件，本规范中固件和 BIOS 不做区分。

**UEFI**：统一的可扩展固定接口，Unified Extensible Firmware Interface，是 Intel 为全新类型的 PC 固件的体系结构、接口和服务提出的建议标准。主要目的是提供在 OS 加载之前在所有平台上一致、正确指定的启动服务，被看做是有近 20 多年历史的 PC BIOS 的继任者。

**PMON**：一种兼有 BIOS 和 boot loader 部分功能的开放源码软件。

**VBIOS**(Video BIOS)：VBIOS 是显卡的 BIOS，VBIOS 提供一些和显示相关的功能，并存放显示芯片与驱动程序之间的控制程序，另外还存放有显示卡型号、规格、生产厂家、出厂时间等信息。

**PCI** (Peripheral Component Interconnect)：是连接电子计算机主板和外部设备的总线标准，用于定义局部总线的标准。此标准允许在计算机内安装多达 10 个遵从 PCI 标准的扩展卡。

**Encoder**：信号编码器，用于将 DVO 信号转换为其他的显示信号，如模拟信号、LVTM 或 TMDS 等显示信号。

**Crtc**：显示控制器，显示控制器把数据从显存中将要显示数据取出来，然后对这些数据做一定的处理送到显示屏。

**Connector**：显示接口连接器，指显示接口硬件，例如 VGA 连接器，HDMI 连接器等。

## 3 架构关系

龙芯 7A1000 桥片中集成显示控制器和 GPU，龙芯 7A1000 显示驱动包括内核驱动和 VBIOS 固件，由内核驱动和 VBIOS 固件共同控制龙芯显示控制器和 GPU，实现显示，背光调节，渲染等功能。VBIOS 固件需要使用 VBIOS 生成工具生成，并存放固定位置(详见第 5 章)，VBIOS 生成工具的使用详见《龙芯 VBIOS1.0 生成工具使用手册》，龙芯 7A1000 内置显卡驱动初始化设备时，从 VBIOS 固件中解析硬件信息，配置，代码等内容。内置显卡内核驱动与 VBIOS 固件层次关系如图 3-1 所示：

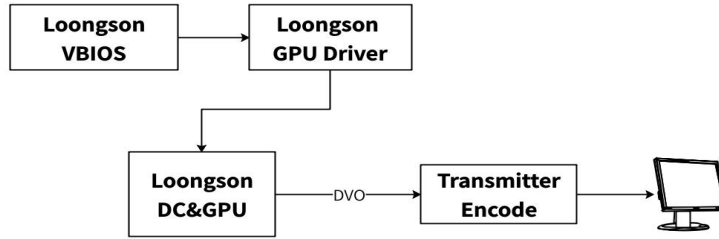


图 3-1 显示驱动，固件与显示硬件的关系

## 4 显示类型和模式

龙芯 7A1000 桥片中集成的显示控制器包含两路 DVO 端口，通过连接外置的编码芯片(Encoder)可实现模拟或者数字信号输出。当前操作系统内核中已集成 LT8618、IT66121、MS7210 三款 HDMI 转换芯片的驱动，正常使用时需由 VBIOS 配合，实现点亮显示器和待机唤醒等功能。

龙芯 7A1000 桥片内置显卡支持的显示模式，如表 4-1 所示。

表 4-1 显示模式支持情况

显示类型	描述
CRT	支持模拟信号显示器(VGA)，通过 DVO 端口连接额外的转换芯片来支持。
显示面板	支持 TMDS(DVI,HDMI)和 LVDS 兼容显示面板,通过 DVO 端口连接额外的转换芯片来支持。

龙芯 7A1000 显示控制器支持多种分辨率，每路显示最大支持 1920x1080@60Hz，支持 RGB565，ARGB8888 两种色深，龙芯显示控制器要求水平分辨满足 256 字节对齐，对齐计算方法：水平宽度 × 色深/8，计算后得到整数则表示对齐，否则表示不对齐，以 1152x864 分辨率 16 位色深为例， $1152 \times 16 / 8 / 256 = 9$ ，1152x864 分辨率是可以正常显示的，在 RGB565 色深模式下，由于多种分辨率无法满足对齐要求，多屏模式下可能出现显示异常，常用分辨率支持情况见表 4-2。

表 4-2 分辨率支持表

分辨率	RGB565	ARGB8888
800x600	不支持	不支持
1024x768	支持	支持
1152x864	支持	支持
1680x1050	不支持	不支持
1280x1024	支持	支持
1600x1200	不支持	支持
1920x1080	支持	支持

## 4.1 多屏显示方案

龙芯 7A1000 显示驱动支持单路和两路同时显示，其中两路显示支持几种不同的模式，支持的情况如表 4-3 所示。

表 4-3 多屏支持情况

显示配置模式	描述	支持情况
Single	支持单独一个显示器	支持
Twin	支持两个显示器，以相同分辨率和时序，显示相同画面，	支持
Clone	支持两个显示器，以不同分辨率和时序，显示相同的画面	不支持
Extended	支持两个显示器，以左右或上下扩展的形式显示完整画面	支持

龙芯 7A1000 显示驱动支持多种硬件链接方案。

方案 A: 硬件上选用一款多通道输出视频编解码转换芯片 (Encoder) 来实现多屏镜像(Twin)显示, 如图 4-1 所示。注意: 该方案只能支持镜像模式的双屏显示。

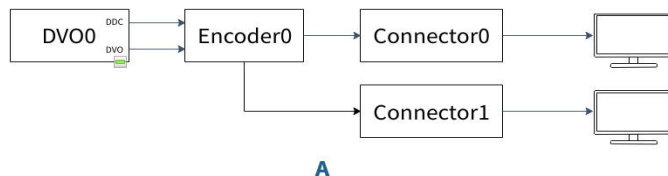


图 4-1 镜像双屏显示硬件链接方案

方案 B: 实现两路独立的双屏显示, 需要同时使用两路显示控制器分别连接相应的视频编解码转换芯片(Encoder)芯片。如图 4-2, 实现多屏扩展(Extended)和多屏镜像(Twin)功能。支持单屏-多屏切换, 多屏-单屏切换和多屏不同模式间的切换。

方案 C: 两路独立的双屏显示, 支持显示控制器(Crtc)和视频编解码转换芯片(Encoder)之间支持交叉连接, 如图 4-2 中, 采用这种方案时 7A1000 显示控制器在硬件设计时需要注意, 同一个显示控制器的 DVO 显示通道和 DDC 通道必须同时连接到一个视频编解码转换芯片(Encoder)。

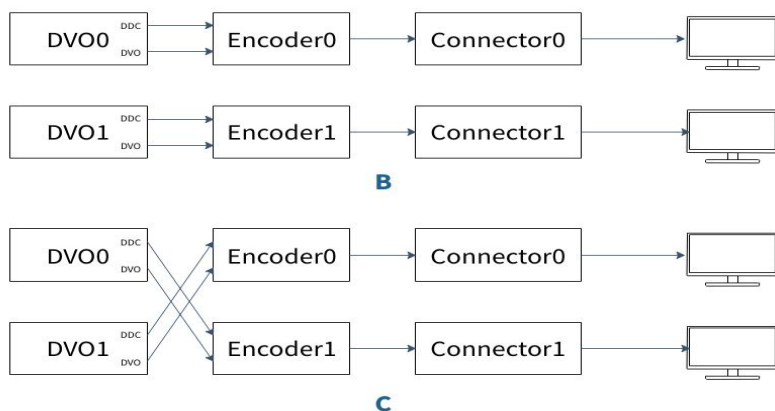


图 4-2 独立双屏显示硬件连接方案

针对方案 B 和方案 C, 显示控制器(Crtc)和视频编解码转换芯片(Encoder)之间的连接关系, 需要通

过 VBIOS 生成工具进行相应的配置，仅在硬件布线正确连接，并且在 VBIOS 中正确配置连接关系后才能正常显示，同时显示接口能获取正确的 EDID 信息，才能实现正常的显示器的拔插检测功能。

## 4.2 DDC 通道硬件连接方案

龙芯 7A1000 桥片对每个显示通路的 DDC 连接方案做了具体规定。龙芯 7A1000 内置显卡支持三种 DDC 通道的连接方案。

方案 A: 使用不需要配置的 Encoder 芯片，则 DVO0\_SCL 和 DVO0\_SDA 经过电平转换后直接连接显示接口，硬件连接方案如图 4-3 所示；

方案 B: 使用需要配置(非透明)的 Encoder 芯片，则 DVO0\_SCL 和 DVO0\_SDA 需要同时作为 Encoder 芯片的配置通道和 EDID 读取通道使用，硬件连接方案如图 4-4 所示；

方案 C: 使用需要配置(非透明)的视频编解码转换芯片(Encoder)，且视频编解码转换芯片(Encoder)支持 EDID 的读取功能,则只需要将 DVO0\_SCL 和 DVO0\_SDA 作为视频编解码转换芯片(Encoder)的配置通道，通过向视频编解码转换芯片(Encoder)发送命令来读取 EDID 信息，如图 4-5 所示。

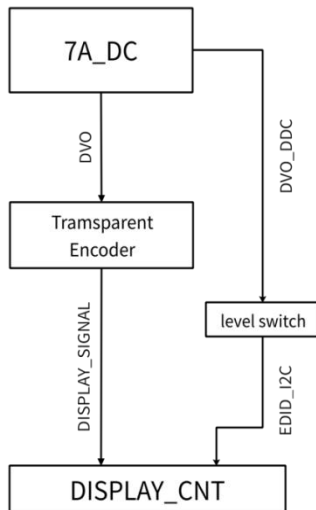


图 4-3 方案 A

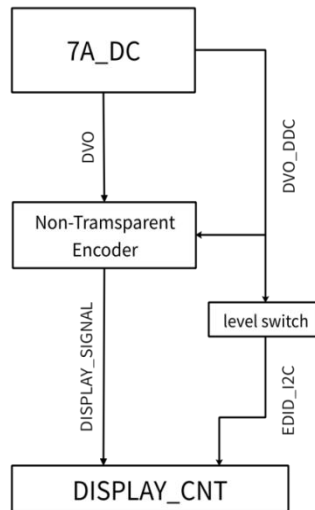


图 4-4 方案 B

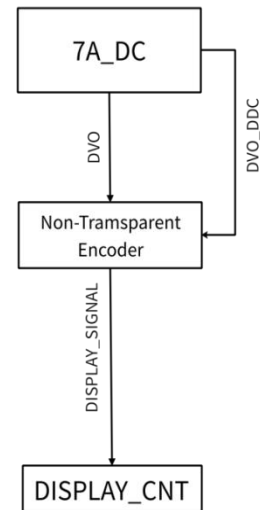


图 4-5 方案 C

## 4.3 获取 EDID

龙芯 7A1000 内置显卡提供两种 EDID 获取方案。

方案 A: 使用显示控制器的 DDC 通道读取外部 EDID 信息，驱动根据不同显示设备，读取不同的 EDID 信息；

方案 B: 在 VBIOS 中预存需要的 EDID 信息，适用于以下三种情况：

1. 外部显示设备无法提供 EDID 信息；
2. 外部显示设备仅能提供一个 EDID 信息，而实际需要使用多组分辨率。
3. 无法采用标准的获取方式获取 EDID。例如，无法通过 i2c 从设备 0x50 地址中读取到 EDID。

## 4.4 热插拔探测

龙芯 7A1000 内置显卡通过软件轮询是否能够获取 EDID 信息来判定是否正常连接显示器,实现热插拔探测,如果能正常读取 EDID 信息则表示显示器连接,否则表示显示器断开。

如需支持热插拔功能,需要按照 4.2 节所述设计硬件,保证驱动能够正确读取到 EDID。如需不支持热插拔功能,需要在 VBIOS 中关闭热插拔探测,关闭热插拔探测情况下,显示接口一直输出显示信号,在这种模式下,如获取不到外部显示设备的 EDID 信息,仅能支持预存储在 VBIOS 或者系统中分辨率。

## 4.5 背光控制

使用 7A1000 内置显卡方案时,硬件设计约束为:显示屏背光开关控制必须使用 7A 桥片上的 GPIO46 和 GPIO47,显示屏背光的亮度控制必须使用 7A 桥片上的 PWM3(GPIO07),如表 4-4 所示,否则背光功能可能无法正常工作。

表 4-4 背光控制硬件链接关系

LCD_EN(LCD backlight enable)	CLKSEL0(GPIO46)默认下拉
LCD_VDD_EN(LCD power enable)	CLKSEL1(GPIO47)默认下拉
LCD_PWM(LCD backlight PWM)	PWM3(GPIO07)

## 5 VBIOS 存储约定

使用制作工具生成的 VBIOS 固件二进制文件存储在 BIOS 同一个 ROM 中。

# 龙芯 7A2000 内置显卡软硬件设计规范

## V1.2

## 版本信息

文档信息	文档名	龙芯 7A2000 内置显卡软硬件设计规范
	版本号	V1.2
	创建人	系统研发部
版本历史		
序号	版本号	更新内容
1	V1.0	V1.0 初始版本
2	V1.1	优化多屏方案的描述
3	V1.2	大幅修订文档内容 文档按照 硬件设计规范和 VBIOS 配置



目录 .....	I
1. 范围 .....	1
2. 术语与定义 .....	1
3. 架构关系 .....	1
4. 7A2000 桥片硬件显示管线设计规范 .....	2
4.1 硬件显示管线简介 .....	2
4.2 显示器显示管线方案 .....	3
4.2.1 显示器显示管线 .....	3
4.2.2 获取 EDID .....	4
4.2.3 热插拔探测 .....	4
4.3 液晶屏显示管线方案 .....	4
4.3.1 液晶屏显示管线 .....	4
4.3.1 获取 EDID .....	5
4.3.2 屏幕背光 .....	5
4.5 显存设计约束 .....	5
4.6 信号转换芯片选型约束 .....	5
4.7 多屏显示方案 .....	6
4.7.1 单屏方案 .....	6

4.7.2 双屏方案 .....	6
4.7.3 三屏方案 .....	6
5. 7A2000 VBIOS 设备属性标配 .....	7
5.1 显示控制器属性表 .....	7
5.2 内置编码器属性表 .....	7
5.3 外接信号转换器属性表 .....	8
5.4 连接器属性表 .....	8
5.5 GPIO 设备属性表 .....	8
5.6 PWM 设备属性表 .....	9
5.7 GPU 属性表 .....	9
5.8 显存属性表 .....	9
6. EFIFB .....	10
7. 附录 .....	10
7.1 显示器方案配置示例 .....	10
7.2 液晶屏方案配置示例 .....	11

## 前 言

本规范是龙芯中科技术有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范涉及到龙芯 7A2000 桥片内置显卡，主要介绍龙芯板卡内置显卡固件与硬件之间，固件和内核接口之间的功能划分及相关硬件布线约定。

# 1 范围

本规范详细介绍龙芯 7A2000 桥片内置显卡的硬件设计约束和 VBIOS 设备属性表。本规范适用于使用龙芯 7A2000 桥片的主板。请其它系统厂商遵循此规范开发相关产品。

# 2 术语与定义

本规范所用术语定义如下：

**固件：**Firmware，写入 ROM、EPROM 等非易失存储器中的程序，负责控制和协调集成电路。

**BIOS：**基本输入输出系统，Basic Input Output System，一组固化到主板上一个 ROM 芯片上的程序，它保存着计算机基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。BIOS 与硬件系统集成在一起，也被称为固件，本规范中固件和 BIOS 不做区分。

**UEFI：**统一的可扩展固定接口，Unified Extensible Firmware Interface，是 Intel 为全新类型的 PC 固件的体系结构、接口和服务提出的建议标准。主要目的是提供在 OS 加载之前在所有平台上一致、正确指定的启动服务，被看做是有近 20 多年历史的 PC BIOS 的继任者。

**PMON：**MIPS 架构机器上使用的一种兼有 BIOS 和 boot loader 部分功能的开放源码软件。

**VBIOS(Video BIOS)：**VBIOS 是显卡的 BIOS，VBIOS 提供一些和显示相关的功能，并存放显示芯片与驱动程序之间的控制程序，另外还存放有显示卡型号、规格、生产厂家、出厂时间等信息。

**PCI (Peripheral Component Interconnect)：**是连接电子计算机主板和外部设备的总线标准，用于定义局部总线的标准。此标准允许在计算机内安装多达 10 个遵从 PCI 标准的扩展卡。

**Encoder：**型号编码器，用于将 DVO 型号转换为其他的显示信号，如模拟型号、LVTM 或 TMDS 等显示信号。

**CRTC：**显示控制器(DC)在驱动中的抽象，显示控制器把数据从显存中将要显示数据取出来，然后对这些数据做一定的处理送到显示屏。

**Connector：**显示接口连接器，指显示接口硬件，例如 VGA 连接器，HDMI 连接器等。

**EDID：**显示器识别数据，存储在显示器中的 DDC 存储器中，当电脑主机与显示器连接后，电脑主机会通过 DDC 通道读取显示器中存储的 EDID。

# 3 架构关系

龙芯 7A2000 桥片中集成显示控制器(DC)和图形处理器(GPU)，龙芯显示驱动包括内核驱动和 VBIOS 固件，GPU 固件。由内核驱动和 VBIOS 固件，GPU 固件共同控制龙芯显示控制器和 GPU，实现显示、背光调节、二三维渲染加速等功能。VBIOS 固件需要由专门的工具制作，龙芯 7A2000 桥片内置 GPU 驱动初始化设备时，从显存中获取 VBIOS 固件，解析 VBIOS 固件中包含的设备属性表，设备配置，代码等内容。龙芯内置显卡驱动与 VBIOS 固件层次关系如图 1-1 所示：

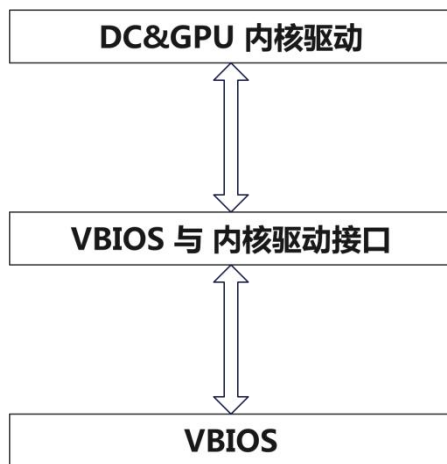


图 1-1 显示驱动，固件与显示硬件的关系

## 4 7A2000 桥片硬件显示管线设计规范

### 4.1 硬件显示管线简介

显示管线是显卡中重要的组成部分,显示管线负责扫描显存中的数据并将这些数据送到终端显示,图形管线由显示控制器,编码器,信号转换器,连接器,I2C 控制器等多种设备组成,显示管线的结构如图 1-1 所示, 每个设备的功能见表 1-1 所示。

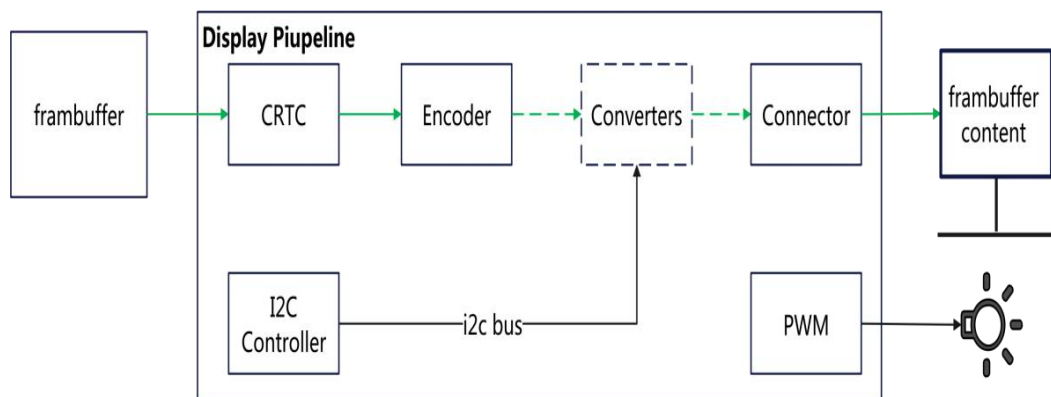


图 1-1 显示管线

设备名称	功能描述
CRTC	显示控制器,对显示缓冲区进行扫描,并产生时序信号的硬件模块。
Encoder	编码器,将显示控制器产生的时序信号转换成外部设备所需要的信号。
Converter	信号转换器,将编码器产生的时序信号转换成外部设备所需要的信号,可选,可支持多个。
Connector	连接器,连接物理显示设备的连接器。
I2C	I2C 控制器,用于控制外部的信号转换器或者通过连接器读取

	EDID。
PWM	PWM 控制器，用于控制外部背光。
VRAM	显存

表 1-1 设备名称功能

本规范约束搭载 7A2000 桥片主板的硬件显示管线设计。主板设计时需要符合本规范中的设计约束，能保证显示驱动软件的兼容性。

## 4.2 显示器显示管线方案

### 4.2.1 显示器显示管线

本规范中定义了两种显示器方案，连接转换芯片和不连接转换芯片两种方案，在设计硬件显示管线方案时可根据具体的需求决定是否使用转换芯片，不同的硬件显示管线方案有不同的约束。

**方案一：**不需要连接信号转换芯片时，硬件设计约束示意图如图 2-1，要求将 7A2000 桥片显示控制器的 I2C 直接连接到显示接口连接器上。用于实现读取 EDID 和使用 VGA 接口时的热插拔探测功能。

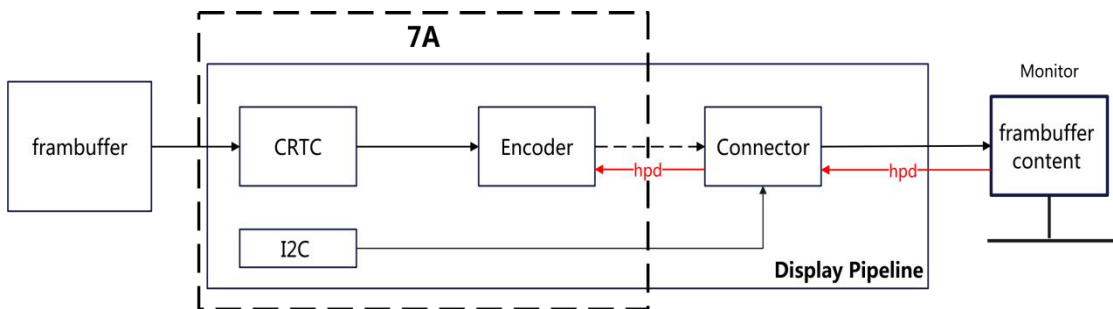


图 2-1 无信号转换芯片的显示器显示管线

**方案二：**需要连接信号转换芯片时，硬件设计约束如图 2-2，要求将 7A2000 桥片的 I2C 连接到转换芯片的 I2C 从接口上，将转换芯片的 I2C 主接口连接到对应的显示接口上。用于通过信号转换芯片实现读取 EDID 和热插拔探测功能。可以使用 7A 桥片上 GPIO52 来控制转换芯片的复位。在有低功耗需求的场景下对转换芯片的功耗进行控制（不强制）。

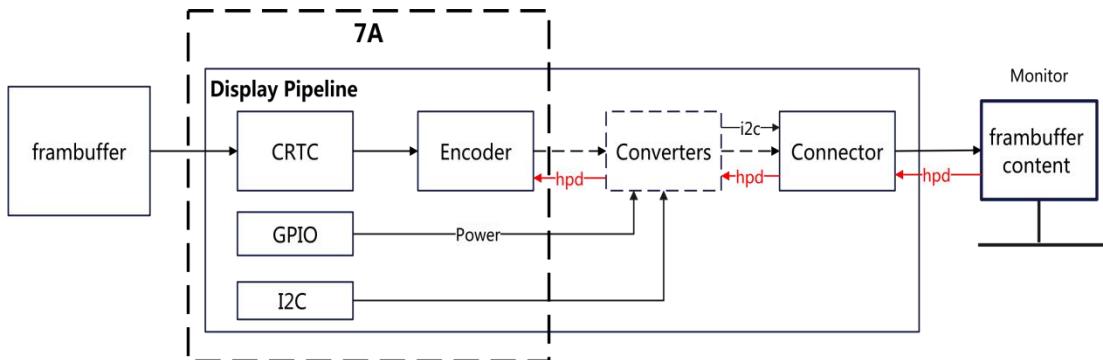


图 2-2 有信号转换芯片的显示器显示管线

## 4.2.2 获取 EDID

在显示器方案中，EDID 由显示器提供。在不连接信号转换芯片时，通过显示控制器中 I2C 直接探测 EDID，在连接信号转换芯片时，通过显示控制器的 I2C 向信号转换芯片发送探测指令，由显示转换芯片完成进一步的探测，并将 EDID 返回给驱动。驱动根据 EDID 中的信息设置显示控制器的输出时序。

## 4.2.3 热插拔探测

在设计显示器显示管线热插拔探测时，请按照如下的原则进行：

1. 直接使用内置 VGA 作为显示接口时，建议使用轮训方式进行热插拔探测，在硬件设计上务必保证具备探测到 EDID，否则热插功能拔会失效。
2. 直接使用内置 HDMI 作为显示接口时，使用中断方式进行热插拔探测。
3. 使用外接转换芯片时，要求连接转换芯片的热插拔中断管脚。

## 4.3 液晶屏显示管线方案

### 4.3.1 液晶屏显示管线

本规范中约束了液晶屏硬件显示管线的实现方案，在液晶屏方案中，支持背光调节，不支持液晶屏热插拔功能。

**方案说明：**将桥片作为信号转换芯片的主控时，硬件参考设计如图 3-1，将显示控制器的 I2C 连接到转换芯片。用于实现对信号转换芯片的控制和读取 EDID 功能。使用桥片的 GPIO 管脚来控制转换视频转换芯片的电源，背光电源和液晶屏电源。

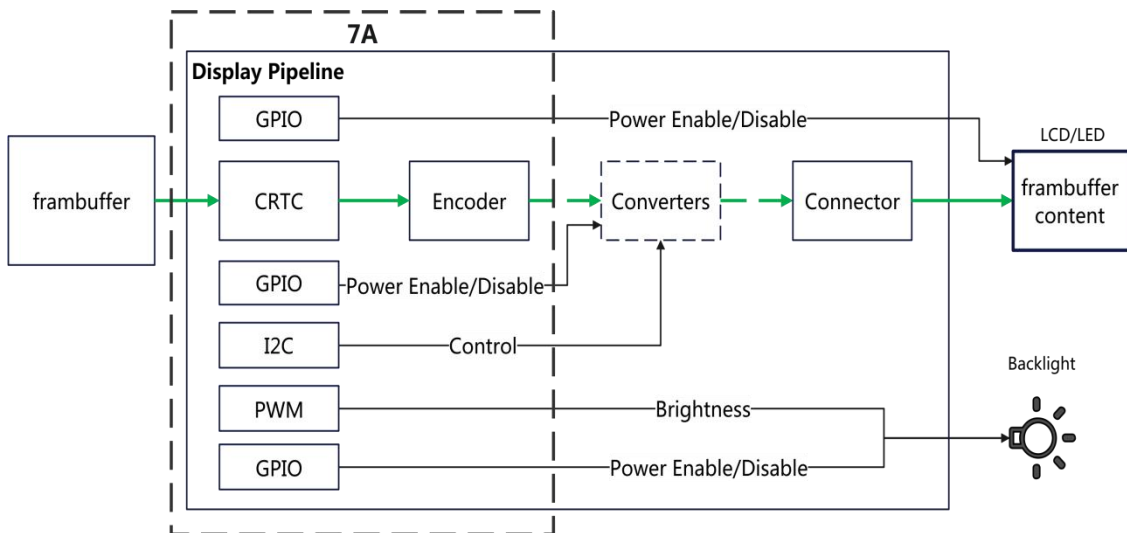


图 3-1 液晶屏显示管线

### 4.3.1 获取 EDID

在液晶屏方案中，EDID 可由液晶屏或者显示转换芯片提供，在前两者无法提供的情况下也可由 VBIOS 提供。EDID 用于提供显示控制器的输出时序参数。EDID 由液晶屏或显示转换芯片提供时，通过 7A2000 桥片显示控制器中的 I2C 向信号转换芯片发送探测指令，这时可由显示转换芯片返回内部存储的 EDID 或者探测液晶屏中存储的 EDID，并将 EDID 返回。如果 EDID 由 VBIOS 提供，则由驱动解析 VBIOS 中的 EDID 数据段。

**需要特别注意的是：EDID 数据应该由液晶屏厂家提供**，厂家在制作 EDID 时，EDID 中的时序必须要符合液晶屏的时序要求，另外 EDID 中的 features 字段(第 0x18 字节)的第 0 位请勿写 1，该位表示显示设备支持工作范围内的 GTF 时序，该位置 1 后驱动将根据 EDID 中提供的时序范围增加 EDID 中未列出的 GTF 时序，使用新增加的分辨率存在不可预测的结果不保证显示功能的正确性和稳定性。关于更多的 EDID 的信息，请参考《VESA ENHANCED EXTENDED DISPLAY IDENTIFICATION DATA STANDARD》

### 4.3.2 屏幕背光

本规范中约束了屏幕背光的实现方案，将桥片作为背光的主控时，由桥片上的 GPIO 和 PWM 分别控制背光电源和背光亮度。推荐使用桥片的 GPIO 和 PWM 资源和状态，见表 4-1。

表 4-1 7A2000 桥片 GPIO，PWM 资源使用表

功能	GPIO 资源与状态
屏幕电源控制	7A2000 GPIO 47，下拉关闭，上拉使能，默认下拉。
屏幕背光控制	7A2000 GPIO 46，下拉关闭，上拉使能，默认下拉。
背光亮度调节	7A2000 PWM 3 (GPIO07)。
信号转换芯片复位控制	7A2000 GPIO 52。

## 4.5 显存设计约束

龙芯 7A2000 桥片支持外接 32 或 16 位宽的显存颗粒，出于对显示性能方面的考虑，推荐使用 32bit 位宽，容量不小于 256 MB。在不接显存颗粒时，7A2000 桥片内置 GPU 将无法正常工作，可能导致操作系统无法正常启动。

## 4.6 信号转换芯片选型约束

对于非透明信号转换芯片选型时需要满足如下的要求：

1. 具备两个 I2C 控制器，作为从设备，用于与主控通信，作为主设备，用于热插拔探测，探测和获取 EDID。
2. 在液晶屏方案下，如需支持多分辨率，要求信号转换芯片支持缩放功能，缩放后的显示质量由转换芯片决定。



3. 具备存储 EDID 的功能(不强制)。
4. 具备低功耗模式。具备电源控制(不强制)。

## 4.7 多屏显示方案

龙芯 7A2000 桥片中有两个显示控制器，最多可支持三路显示，其中 HDMI0 和 VGA 复用相同的信号源，这两路显示内容相同，HDMI1 为独立的信号源，可与 HDMI0 或 VGA 组成组成扩展显示模式。

### 4.7.1 单屏方案

在单屏方案中，可选择单独使用 HDMI0 或者 VGA，也可单独使用 HDMI1。如果使用 VGA 接口时使用轮训作为热插拔探测方案。使用 HDMI 时使用中断作为热插拔探测方案。

### 4.7.2 双屏方案

在双屏方案中，可使用 HDMI0 与 HDMI1 组成双屏方案，可使用 VGA 与 HDMI1 组成双屏方案，采用这两种方案时，两个屏幕可显示不同的内容，实现扩展显示功能；如使用 HDMI0 和 VGA 组成双屏方案时，两个屏幕只能显示相同的内容，且必须使用相同品牌和型号的显示器，否则存在无法正确显示的风险。

### 4.7.3 三屏方案

在三屏方案中，需要同时使用 VGA，HDMI0，HDMI1 组成三屏方案，由于 VGA 和 HDMI0 是相同的信号源，这两路必须使用相同品牌，相同型号的显示器，否则存在无法正确显示的风险。出现不能正确显示的问题时，符合预期。在硬件设计时需要增加如下的开关电路保证 VGA 和 HDMI0 都能探测到 EDID，在三屏方案时 VGA 和 HDMI0 需要热插拔探测(HPD)对多路复用电路 (MUX) 开关进行切换，在 VBIOS 配置方面，VGA 和 HDMI0 的热插拔探测方法需要配置为轮询。

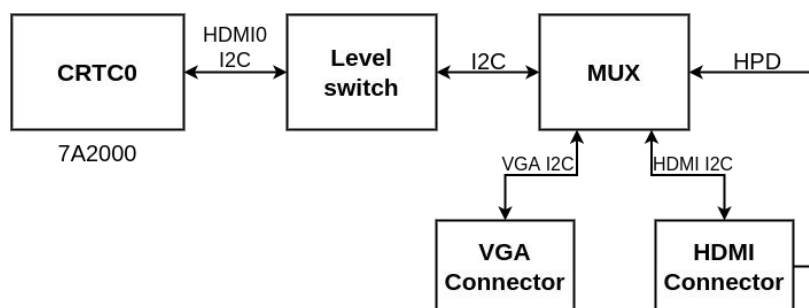


图 1-1 热插拔多路复用电路示意图

## 5 7A2000 VBIOS 设备属性标配

龙芯 VBIOS 提供一种结构化的数据，用于描述无法被 GPU 驱动程序动态探测的设备属性和设备连接拓扑，如图 5-1 所示，通过固件传参的形式将这些设备属性表传递给内核。这些设备可以在 GPU 内部或直接与 GPU 相连。关于 VBIOS 的更多信息，详见参考《龙芯 VBIOS 规范 V2.0》文档，VBIOS 生成工具的使用方法详见集成 VBIOS 生成工具中的《龙芯 VBIOS 生成工具使用手册》。

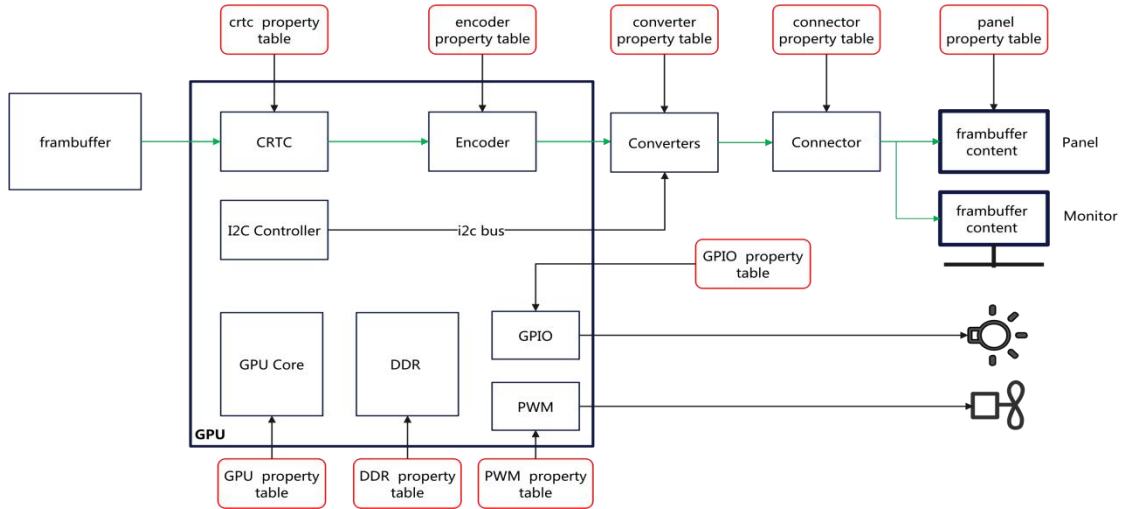


图 5-1 设备属性表示意图

### 5.1 显示控制器属性表

表 5-1 显示控制器属性表

属性名称	类型(大小)	功能说明
feature	u32(4B)	设备特性标记位。
crtc_id	u32(4B)	显示控制器硬件编号。编号从 0 开始。
encoder_id	u32(4B)	与该显示控制器相连的内置信号编码器编号。
max_freq	u32(4B)	控制器硬件支持的最大像素时钟频率，参考芯片手册填写。
max_width	u32(4B)	控制器硬件能支持的最大分辨率宽度，参考芯片手册填写。
max_height	u32(4B)	控制器硬件能支持的最大分辨率高度，参考芯片手册填写。
is_vb_timing	u8(4B)	是否由 vbios 提供显示控制器的时序，默认值为 0，已废弃，工具中不提供配置界面。

### 5.2 内置编码器属性表

表 5-2 信号转换器属性表

属性说明	类型(大小)	说明
feature	u32(4B)	设备特性标记位。

type	u32(4B)	内置编码器输出信号类型。
connector_id	u32(4B)	显示链路编号。

## 5.3 外接信号转换器属性表

表 5-3 内置编码器属性表

属性名称	类型(大小)	功能说明
feature	u32(4B)	设备特性标记位。
i2c_id	u32(4B)	与转换芯片相连的 i2c 编号。 <b>默认值为 -1，已废弃。工具中不提供配置界面。</b>
connector_id	u32(4B)	显示链路编号。
type	u32(4B)	外接编码器芯片输出信号的类型
config_type	u32(4B)	编码器芯片配置方式。
chip_id	u32(4B)	外接转换芯片的型号。
chip_addr	u32(4B)	外接转换芯片 i2c 从设备地址。

## 5.4 连接器属性表

表 5-4 接口连接器属性表

属性名称	类型(大小)	功能说明
feature	u32(4B)	设备特性标记位。
i2c_id	u32(4B)	与连接器相连 i2c 编号。 <b>默认值为 -1，已废弃，工具中不提供配置界面。</b>
internal_edid	u32(256B)	存放由 vbiOS 提供的 edid。
type	u32(4B)	连接器接口类型。
hotplug	u32(4B)	热插拔探测方式的类型。
edid_method	u32(4B)	获取 EDID 方式。
irq_gpio	u32(4B)	用于热插拔探测的 GPIO 号。 <b>7A2000 以后不再使用。工具中不提供配置界面。</b>
gpio_placement	u32(4B)	使用 GPIO 的位置， <b>默认值为 7A，已废弃，工具中不提供配置界面。</b>

## 5.5 GPIO 设备属性表

表 5-6 GPIO 设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位

gpio_id	u32(4B)	使用的 gpio 编号
usage	u32(4B)	gpio 用途

## 5.6 PWM 设备属性表

表 5-7 PWM 设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位。
pwm_id	u32(4B)	pwm 设备编号，参考主板进行配置。
polarity	u32(4B)	pwm 设备极性配置，参考主板进行配置。
peroid	u32(4B)	pwm 设备周期，单位 Hz，参考芯片手册进行配置。
usage	u32(4B)	pwm 控制器的用途，参考主板进行配置。

## 5.7 GPU 属性表

表 5-8 GPU 设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位
gpu_freq_level	u32(4B)	GPU 频率调节档位，参考硬件配置。
gpu_freq	u32(4B)	GPU 最大频率，参考硬件配置。
num_gpu_shaders	u32(4B)	GPU 流处理器个数，参考硬件配置。

## 5.8 显存属性表

表 5-9 显存设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位。
vram_type	u32(4B)	显存颗粒类型。参考主板设计配置。
vram_bit_width	u32(4B)	显存颗粒位宽。参考主板设计配置。
vram_capacity	u32(4B)	显存容量。参考主板设计配置。
vram_freq	u32(4B)	显存频率。参考主板设计配置。

## 6 efifb

efifb 适用于符合 UEFI 规范的固件。efifb 既支持带有图形输出协议 (GOP) 显示器的固件。efifb 可以复用固件启动时的分辨率和帧缓冲区地址。因此可以实现在操作系统没有显示驱动的情况下支持基本的图形显示功能。

龙芯 BIOS 中已默认集成支持 GOP 驱动，在操作系统中没有集成显卡驱动或不满足显卡驱动运行条件时，可继续由 efifb 完成图形显示功能。**要求固件传递帧缓冲地址，分辨率，色深必须和固件使用相同，才能保证进入系统后正常显示。**

系统使用 efifb 进入桌面后可正常操作应用程序，如开启浏览器、终端、文件管理等应用程序；efifb 只能提供基础的图形显示功能，不支持画面旋转、切换分辨率、更改刷新率、显示器热插拔、显示器信息获取等功能。为使操作系统获得完整的使用体验，使用 efifb 启动后请尝试安装最新显卡驱动。

## 7 附录

### 7.1 显示器方案配置示例

如表 7-1 和表 7-2 所示，为使用内置 HDMI 直接输出显示器时 vbios 中设备信息的描述表。

表 7-1 内置 HDMI 配置

设备	显示控制器	内置编码器	外接编码器	连接器
属性/配置	feature (0)	feature (0)	feature (0)	feature (0)
	crtc_id (0)	encoder_id (0)	i2c_id (0)	i2c_id (0)
	encoder_id (0)	connector_id (0)	connector_id (0)	internal_edid (不配置)
	max_freq (340000)		type (TMDS)	type (HDMI)
	max_width (4096)		config_type (driver)	hotplug (irq)
	max_height (4096)		chip_id (INTERNA_HDMI)	edid_method (I2C)
			chip_addr (0)	

表 7-2 GPU 配置

设备	GPU	显存
属性/配置	feature (0)	feature (0)
	gpu_freq_level (3)	vram_type (ddr4)
	gpu_freq (500MHz)	vram_bit_width (32 bit)
	num_gpu_shaders (64)	vram_capacity (1GB)
		vram_freq (2400MHz)

## 7.2 液晶屏方案配置示例

如表 7-3 和表 7-4 所示，为使用外接 eDP 转换芯片输出液晶屏时 vbios 对各设备信息的描述。

表 7-3 外接 eDP 配置

设备	显示控制器	内置编码器	外接编码器	连接器	PWM
属性/配置	feature (0)	feature (0)	feature (0)	feature (0)	feature (0)
	crtc_id (0)	encoder_id (0)	i2c_id (0)	i2c_id (0)	pwm_id (3)
	encoder_id (0)	connector_id (0)	connector_id (0)	internal_edid (不配置)	polarity (1000000)
	max_freq (340000)		type (LVDS)	type (eDP)	peroid (1)
	max_width (4096)		config_type (driver)	hotplug (connected)	usage (0)
	max_height (4096)		chip_id (lt9721)	edid_method (I2C)	
			chip_addr (0x3a)		

表 7-4 GPU 配置

设备	GPU	显存
属性/配置	feature (0)	feature (0)
	gpu_freq_level (3)	vram_type (ddr4)
	gpu_freq (500MHz)	vram_bit_width (32 bit)
	num_gpu_shaders (64)	vram_capacity (1 GB)
		vram_freq (2400MHz)

# 龙芯 2K2000 内置显卡软硬件设计规范

## V1.0

## 版本信息

文档信息	文档名	龙芯 7A2000 内置显卡软硬件设计规范
	版本号	V1.0
	创建人	系统研发部
版本历史		
序号	版本号	更新内容
1	V1.0	V1.0 初始版本



1. 范围 .....	1
2. 术语与定义 .....	1
3. 架构关系 .....	1
4. 2K2000 芯片硬件显示管线设计规范 .....	2
4.1 硬件显示管线简介 .....	2
4.2 显示器显示管线方案 .....	3
4.2.1 显示器显示管线 .....	3
4.2.2 获取 EDID .....	4
4.2.3 热插拔探测 .....	4
4.3 液晶屏显示管线方案 .....	4
4.3.1 液晶屏显示管线 .....	4
4.3.1 获取 EDID .....	5
4.3.2 屏幕背光 .....	5
4.5 显存设计约束 .....	5
4.6 信号转换芯片选型约束 .....	5
4.7 多屏显示方案 .....	6
4.7.1 单屏方案 .....	6
4.7.2 双屏方案 .....	6

5. 2K2000 VBIOS 设备属性标配 .....	6
5.1 显示控制器属性表 .....	7
5.2 内置编码器属性表 .....	7
5.3 外接信号转换器属性表 .....	7
5.4 连接器属性表 .....	8
5.5 GPIO 设备属性表 .....	8
5.6 PWM 设备属性表 .....	8
5.7 GPU 属性表 .....	8
5.8 显存属性表 .....	9
6. EFIFB .....	9
7. 附录 .....	9
7.1 显示器方案配置示例 .....	9
7.2 液晶屏方案配置示例 .....	10

## 前 言

本规范是龙芯中科技术有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范涉及到龙芯 2K2000 芯片内置显卡，主要介绍龙芯板卡内置显卡固件与硬件之间，固件和内核接口之间的功能划分及相关硬件布线约定。

# 1 范围

本规范详细介绍龙芯 2K2000 芯片内置显卡的硬件设计约束和 VBIOS 设备属性表。本规范适用于使用龙芯 2K2000 芯片的主板。请其它系统厂商遵循此规范开发相关产品。

# 2 术语与定义

本规范所用术语定义如下：

**固件：**Firmware，写入 ROM、EPROM 等非易失存储器中的程序，负责控制和协调集成电路。

**BIOS：**基本输入输出系统，Basic Input Output System，一组固化到主板上一个 ROM 芯片上的程序，它保存着计算机基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。BIOS 与硬件系统集成在一起，也被称为固件，本规范中固件和 BIOS 不做区分。

**UEFI：**统一的可扩展固定接口，Unified Extensible Firmware Interface，是 Intel 为全新类型的 PC 固件的体系结构、接口和服务提出的建议标准。主要目的是提供在 OS 加载之前在所有平台上一致、正确指定的启动服务，被看做是有近 20 多年历史的 PC BIOS 的继任者。

**PMON：**MIPS 架构机器上使用的一种兼有 BIOS 和 boot loader 部分功能的开放源码软件。

**VBIOS(Video BIOS)：**VBIOS 是显卡的 BIOS，VBIOS 提供一些和显示相关的功能，并存放显示芯片与驱动程序之间的控制程序，另外还存放有显卡型号、规格、生产厂家、出厂时间等信息。

**PCI (Peripheral Component Interconnect)：**是连接电子计算机主板和外部设备的总线标准，用于定义局部总线的标准。此标准允许在计算机内安装多达 10 个遵从 PCI 标准的扩展卡。

**Encoder：**型号编码器，用于将 DVO 型号转换为其他的显示信号，如模拟型号、LVTM 或 TMDS 等显示信号。

**CRTC：**显示控制器(DC)在驱动中的抽象，显示控制器把数据从显存中将要显示数据取出来，然后对这些数据做一定的处理送到显示屏。

**Connector：**显示接口连接器，指显示接口硬件，例如 VGA 连接器，HDMI 连接器等。

**EDID：**显示器识别数据，存储在显示器中的 DDC 存储器中，当电脑主机与显示器连接后，电脑主机会通过 DDC 通道读取显示器中存储的 EDID。

# 3 架构关系

龙芯 2K2000 芯片中集成显示控制器(DC)和图形处理器(GPU)，龙芯显示驱动包括内核驱动和 VBIOS 固件，GPU 固件。由内核驱动和 VBIOS 固件，GPU 固件共同控制龙芯显示控制器和 GPU，实现显示、背光调节、二三维渲染加速等功能。VBIOS 固件需要由专门的工具制作，龙芯 2K2000 芯片内置 GPU 驱动初始化设备时，从显存中获取 VBIOS 固件，解析 VBIOS 固件中包含的设备属性表，设备配置，代码等内容。龙芯内置显卡驱动与 VBIOS 固件层次关系如图 1-1 所示：

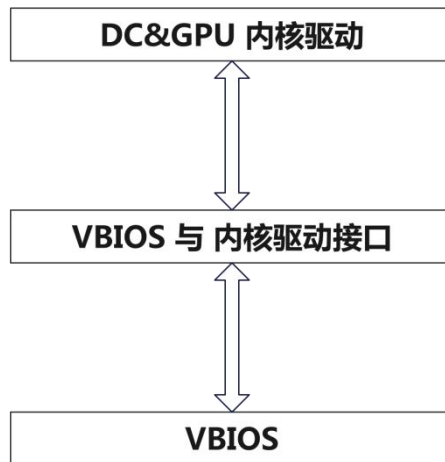


图 1-1 显示驱动，固件与显示硬件的关系

## 4 2K2000 芯片硬件显示管线设计规范

### 4.1 硬件显示管线简介

显示管线是显卡中重要的组成部分,显示管线负责扫描显存中的数据并将这些数据送到终端显示,图形管线由显示控制器,编码器,信号转换器,连接器,I2C 控制器等多种设备组成,显示管线的结构如图 1-1 所示, 每个设备的功能见表 1-1 所示。

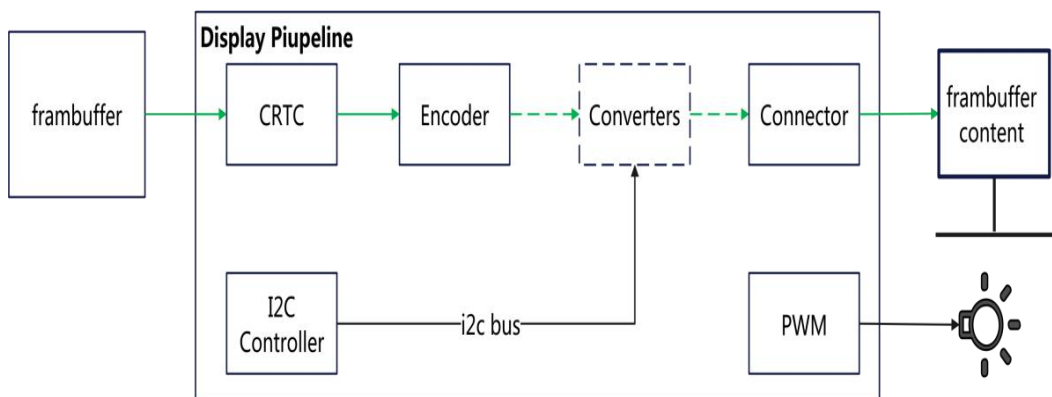


图 1-1 显示管线

设备名称	功能描述
CRTC	显示控制器,对显示缓冲区进行扫描,并产生时序信号的硬件模块。
Encoder	编码器,将显示控制器产生的时序信号转换成外部设备所需要的信号。
Converter	信号转换器,将编码器产生的时序信号转换成外部设备所需要的信号,可选,可支持多个。
Connector	连接器,连接物理显示设备的连接器。
I2C	I2C 控制器,用于控制外部的信号转换器或者通过连接器读取

	EDID。
PWM	PWM 控制器，用于控制外部背光。
VRAM	显存

表 1-1 设备名称功能

本规范约束搭载 2K2000 芯片主板的硬件显示管线设计。主板设计时需要符合本规范中的设计约束，能保证显示驱动软件的兼容性。

## 4.2 显示器显示管线方案

### 4.2.1 显示器显示管线

本规范中定义了两种显示器方案，连接转换芯片和不连接转换芯片两种方案，在设计硬件显示管线方案时可根据具体的需求决定是否使用转换芯片，不同的硬件显示管线方案有不同的约束。

**方案一：**不需要连接信号转换芯片时，硬件设计约束示意图如图 2-1，要求将 2K2000 芯片显示控制器的 I2C 直接连接到显示接口连接器上。用于实现读取 EDID 和使用 VGA 接口时的热插拔探测功能。

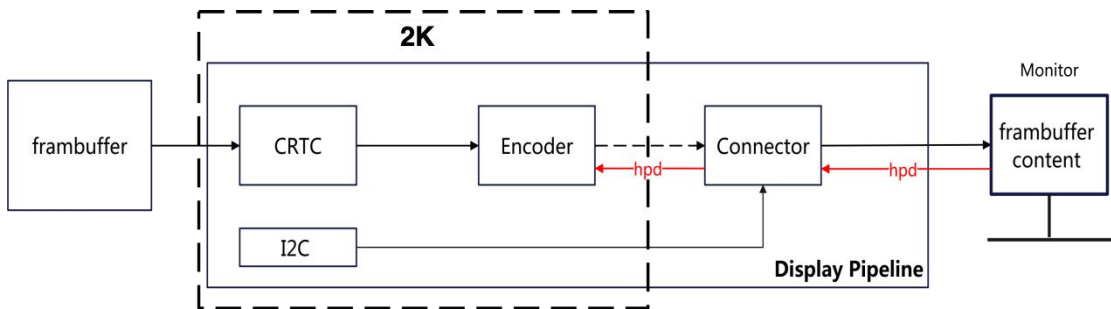


图 2-1 无信号转换芯片的显示器显示管线

**方案二：**需要连接信号转换芯片时，硬件设计约束如图 2-2，要求将 2K2000 芯片的 I2C 连接到转换芯片的 I2C 从接口上，将转换芯片的 I2C 主接口连接到对应的显示接口上。用于通过信号转换芯片实现读取 EDID 和热插拔探测功能。可以使用 2K 芯片上 node GPIO02 来控制转换芯片的复位。在有低功耗需求的场景下对转换芯片的功耗进行控制（不强制）。

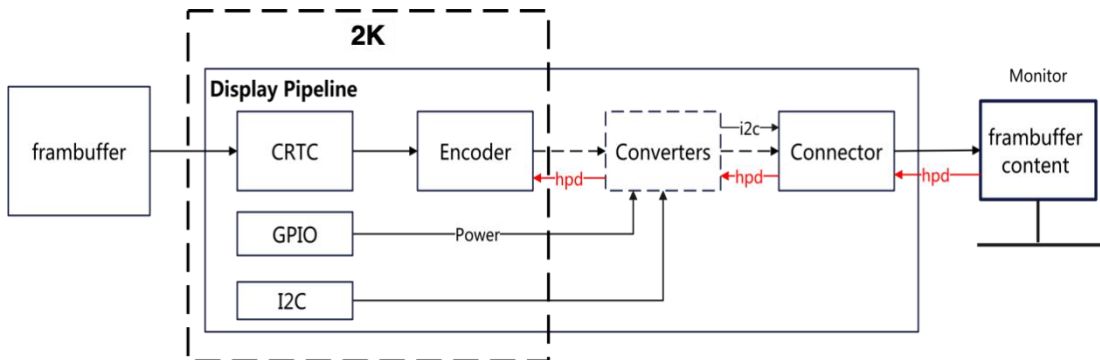


图 2-2 有信号转换芯片的显示器显示管线

## 4.2.2 获取 EDID

在显示器方案中，EDID 由显示器提供。在不连接信号转换芯片时，通过显示控制器中 I2C 直接探测 EDID，在连接信号转换芯片时，通过显示控制器的 I2C 向信号转换芯片发送探测指令，由显示转换芯片完成进一步的探测，并将 EDID 返回给驱动。驱动根据 EDID 中的信息设置显示控制器的输出时序。

## 4.2.3 热插拔探测

在设计显示器显示管线热插拔探测时，请按照如下的原则进行：

4. 直接使用内置 VGA 作为显示接口时，建议使用轮训方式进行热插拔探测，在硬件设计上务必保证具备探测到 EDID，否则热插功能拔会失效。
5. 直接使用内置 HDMI 作为显示接口时，使用中断方式进行热插拔探测。
6. 使用外接转换芯片时，要求连接转换芯片的热插拔中断管脚。

## 4.3 液晶屏显示管线方案

### 4.3.1 液晶屏显示管线

本规范中约束了液晶屏硬件显示管线的实现方案，在液晶屏方案中，支持背光调节，不支持液晶屏热插拔功能。

**方案说明：**将 2K2000 作为信号转换芯片的主控时，硬件参考设计如图 3-1，将显示控制器的 I2C 连接到转换芯片。用于实现对信号转换芯片的控制和读取 EDID 功能。使用芯片的 GPIO 管脚来控制转换视频转换芯片的电源，背光电源和液晶屏电源。

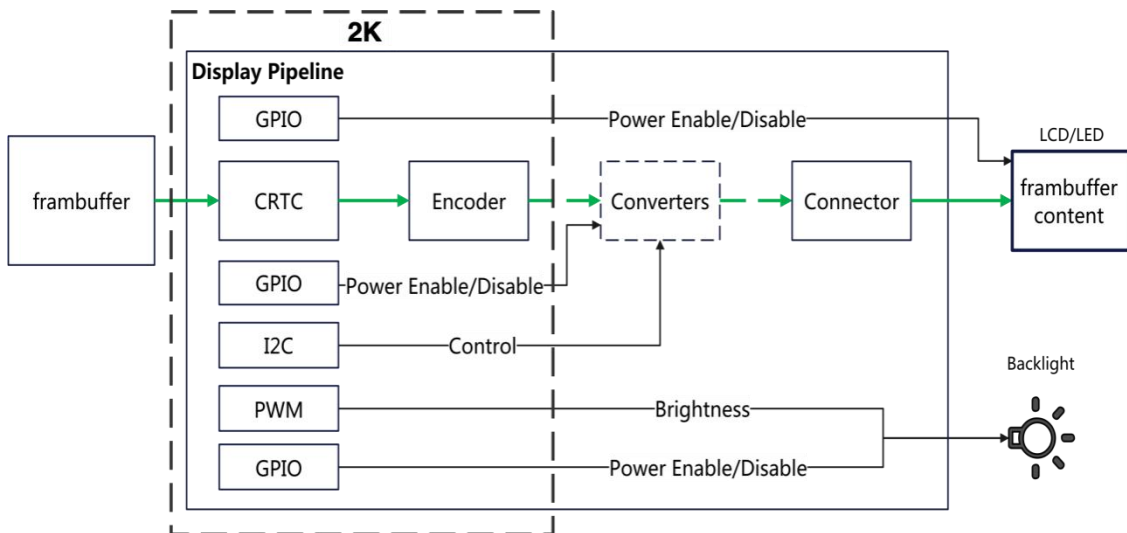


图 3-1 液晶屏显示管线

### 4.3.1 获取 EDID

在液晶屏方案中，EDID 可由液晶屏或者显示转换芯片提供，在前两者无法提供的情况下也可由 VBIOS 提供。EDID 用于提供显示控制器的输出时序参数。EDID 由液晶屏或显示转换芯片提供时，通过 2K2000 芯片显示控制器中的 I2C 向信号转换芯片发送探测指令，这时可由显示转换芯片返回内部存储的 EDID 或者探测液晶屏中存储的 EDID，并将 EDID 返回。如果 EDID 由 VBIOS 提供，则由驱动解析 VBIOS 中的 EDID 数据段。

**需要特别注意的是：EDID 数据应该由液晶屏厂家提供**，厂家在制作 EDID 时，EDID 中的时序必须要符合液晶屏的时序要求，另外 EDID 中的 features 字段(第 0x18 字节)的第 0 位请勿写 1，该位表示显示设备支持工作范围内的 GTF 时序，该位置 1 后驱动将根据 EDID 中提供的时序范围增加 EDID 中未列出的 GTF 时序，使用新增加的分辨率存在不可预测的结果不保证显示功能的正确性和稳定性。关于更多的 EDID 的信息，请参考《VESA ENHANCED EXTENDED DISPLAY IDENTIFICATION DATA STANDARD》

### 4.3.2 屏幕背光

本规范中约束了屏幕背光的实现方案，将芯片作为背光的主控时，由芯片上的 GPIO 和 PWM 分别控制背光电源和背光亮度。推荐使用芯片的 GPIO 和 PWM 资源和状态，见表 4-1。

表 4-1 2K2000 芯片 GPIO ， PWM 资源使用表

功能	GPIO 资源与状态
屏幕电源控制	2K2000 PWM 05 (GPIO 09)，下拉关闭，上拉使能，默认下拉。
屏幕背光控制	2K2000 PWM 04 (GPIO 08)，下拉关闭，上拉使能，默认下拉。
背光亮度调节	2K2000 PWM 03 (GPIO07)。
信号转换芯片复位控制	2K2000 node GPIO 02。

## 4.5 显存设计约束

龙芯 2K2000 芯片支持使用系统内存作为显存，容量不小于 256MB。

## 4.6 信号转换芯片选型约束

对于非透明信号转换芯片选型时需要满足如下的要求：

1. 具备两个 I2C 控制器，作为从设备，用于与主控通信，作为主设备，用于热插拔探测，探测和获取 EDID。
2. 在液晶屏方案下，如需支持多分辨率，要求信号转换芯片支持缩放功能，缩放后的显示质量由转换芯片决定。
3. 具备存储 EDID 的功能(不强制)。



4. 具备低功耗模式。具备电源控制(不强制)。

## 4.7 多屏显示方案

龙芯 2K2000 芯片中有两个显示控制器,最多可支持双路显示,可由 HDMI(显示通道 0)和 DVO (显示通道 1) 组成扩展显示模式。

### 4.7.1 单屏方案

在单屏方案中,可选择单独使用 HDMI (显示通道 0) 或者 DVO (显示通道 1)。

### 4.7.2 双屏方案

在双屏方案中,可使用 HDMI 与 DVO 组成双屏方案,两个屏幕可显示不同的内容,实现扩展显示功能。

## 5 2K2000 VBIOS 设备属性标配

龙芯 VBIOS 提供一种结构化的数据,用于描述无法被 GPU 驱动程序动态探测的设备属性和设备连接拓扑,如图 5-1 所示,通过固件传参的形式将这些设备属性表传递给内核。这些设备可以在 GPU 内部或直接与 GPU 相连。关于 VBIOS 的更多信息,详见参考《龙芯 VBIOS 规范 V2.0》文档, VBIOS 生成工具的使用方法详见集成 VBIOS 生成工具中的《龙芯 VBIOS 生成工具使用手册》。

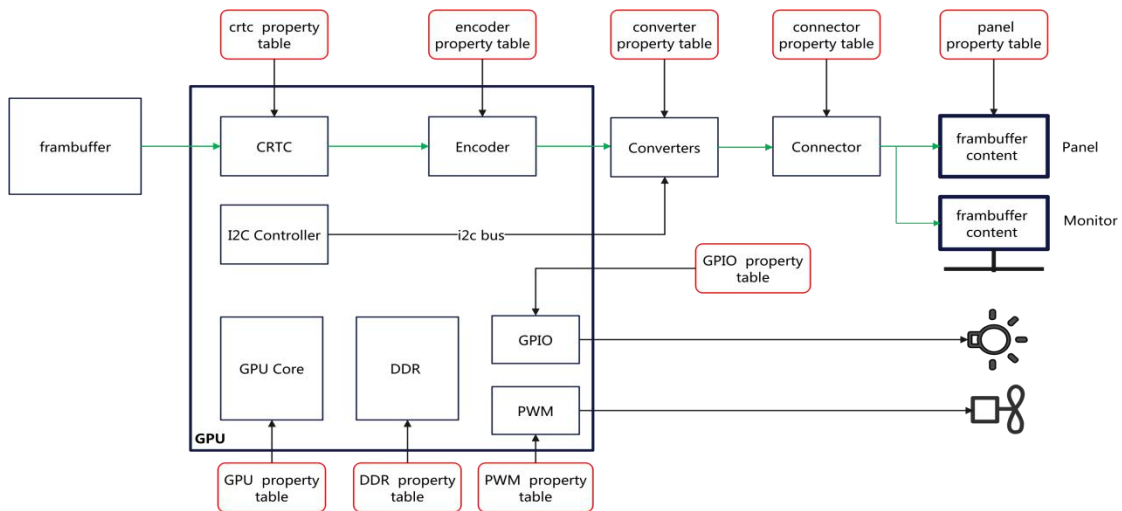


图 5-1 设备属性表示意图

## 5.1 显示控制器属性表

表 5-1 显示控制器属性表

属性名称	类型(大小)	功能说明
feature	u32(4B)	设备特性标记位。
crtc_id	u32(4B)	显示控制器硬件编号。编号从 0 开始。
encoder_id	u32(4B)	与该显示控制器相连的内置信号编码器编号。
max_freq	u32(4B)	控制器硬件支持的最大像素时钟频率，参考芯片手册填写。
max_width	u32(4B)	控制器硬件能支持的最大分辨率宽度，参考芯片手册填写。
max_height	u32(4B)	控制器硬件能支持的最大分辨率高度，参考芯片手册填写。
is_vb_timing	u8(4B)	是否由 vbios 提供显示控制器的时序，默认值为 0，已废弃，工具中不提供配置界面。

## 5.2 内置编码器属性表

表 5-2 信号转换器属性表

属性说明	类型(大小)	说明
feature	u32(4B)	设备特性标记位。
type	u32(4B)	内置编码器输出信号类型。
connector_id	u32(4B)	显示链路编号。

## 5.3 外接信号转换器属性表

表 5-3 内置编码器属性表

属性名称	类型(大小)	功能说明
feature	u32(4B)	设备特性标记位。
i2c_id	u32(4B)	与转换芯片相连的 i2c 编号。默认值为 -1，已废弃。工具中不提供配置界面。
connector_id	u32(4B)	显示链路编号。
type	u32(4B)	外接编码器芯片输出信号的类型
config_type	u32(4B)	编码器芯片配置方式。
chip_id	u32(4B)	外接转换芯片的型号。
chip_addr	u32(4B)	外接转换芯片 i2c 从设备地址。

## 5.4 连接器属性表

表 5-4 接口连接器属性表

属性名称	类型(大小)	功能说明
feature	u32(4B)	设备特性标记位。
i2c_id	u32(4B)	与连接器相连 i2c 编号。 <b>默认值为 -1，已废弃，工具中不提供配置界面。</b>
internal_edid	u32(256B)	存放由 vbios 提供的 edid。
type	u32(4B)	连接器接口类型。
hotplug	u32(4B)	热插拔探测方式的类型。
edid_method	u32(4B)	获取 EDID 方式。
irq_gpio	u32(4B)	用于热插拔探测的 GPIO 号。2K2000 以后不再使用。 <b>工具中不提供配置界面。</b>
gpio_placement	u32(4B)	使用 GPIO 的位置， <b>默认值为 2K，已废弃，工具中不提供配置界面。</b>

## 5.5 GPIO 设备属性表

表 5-6 GPIO 设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位
gpio_id	u32(4B)	使用的 gpio 编号
usage	u32(4B)	gpio 用途

## 5.6 PWM 设备属性表

表 5-7 PWM 设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位。
pwm_id	u32(4B)	pwm 设备编号，参考主板进行配置。
polarity	u32(4B)	pwm 设备极性配置，参考主板进行配置。
peroid	u32(4B)	pwm 设备周期，单位 Hz，参考芯片手册进行配置。
usage	u32(4B)	pwm 控制器的用途，参考主板进行配置。

## 5.7 GPU 属性表

表 5-8 GPU 设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位
gpu_freq_level	u32(4B)	GPU 频率调节档位，参考硬件配置。
gpu_freq	u32(4B)	GPU 最大频率，参考硬件配置。
num_gpu_shaders	u32(4B)	GPU 流处理器个数，参考硬件配置。

## 5.8 显存属性表

表 5-9 显存设备属性表

属性名称	类型(大小)	说明
feature	u32(4B)	设备特性标记位。
vram_type	u32(4B)	显存颗粒类型。参考主板设计配置。
vram_bit_width	u32(4B)	显存颗粒位宽。参考主板设计配置。
vram_capacity	u32(4B)	显存容量。参考主板设计配置。
vram_freq	u32(4B)	显存频率。参考主板设计配置。

## 6 efifb

efifb 适用于符合 UEFI 规范的固件。efifb 既支持带有图形输出协议（GOP）显示器的固件。efifb 可以复用固件启动时的分辨率和帧缓冲区地址。因此可以在操作系统没有显示驱动的情况下支持基本的图形显示功能。

龙芯 BIOS 中已默认集成支持 GOP 驱动，在操作系统中没有集成显卡驱动或不满足显卡驱动运行条件时，可继续由 efifb 完成图形显示功能。**要求固件传递帧缓冲地址，分辨率，色深必须和固件使用相同，才能保证进入系统后正常显示。**

系统使用 efifb 进入桌面后可正常操作应用程序，如开启浏览器、终端、文件管理等应用程序；efifb 只能提供基础的图形显示功能，不支持画面旋转、切换分辨率、更改刷新率、显示器热插拔、显示器信息获取等功能。为使操作系统获得完整的使用体验，使用 efifb 启动后请尝试安装最新显卡驱动。

## 7 附录

### 7.1 显示器方案配置示例

如表 7-1 和表 7-2 所示，为使用内置 HDMI 直接输出显示器时 vbios 中设备信息的描述表。

表 7-1 内置 HDMI 配置

设备	显示控制器	内置编码器	外接编码器	连接器
	feature (0)	feature (0)	feature (0)	feature (0)

属性/配置	crtc_id (0)	encoder_id (0)	i2c_id (0)	i2c_id (0)
	encoder_id (0)	connector_id (0)	connector_id (0)	internal_edid (不配置)
	max_freq (340000)		type (TMDS)	type (HDMI)
	max_width (4096)		config_type (driver)	hotplug (irq)
	max_height (4096)		chip_id (INTERNA_HDMI)	edid_method (I2C)
			chip_addr (0)	

表 7-2 GPU 配置

设备	GPU	显存
属性/配置	feature (0)	feature (0)
	gpu_freq_level (3)	vram_type (ddr4)
	gpu_freq (500MHz)	vram_bit_width (32 bit)
	num_gpu_shaders (64)	vram_capacity (1GB)
		vram_freq (2400MHz)

## 7.2 液晶屏方案配置示例

如表 7-3 和表 7-4 所示，为使用外接 eDP 转换芯片输出液晶屏时 vbios 对各设备信息的描述。

表 7-3 外接 eDP 配置

设备	显示控制器	内置编码器	外接编码器	连接器	PWM
属性/配置	feature (0)	feature (0)	feature (0)	feature (0)	feature (0)
	crtc_id (0)	encoder_id (0)	i2c_id (0)	i2c_id (0)	pwm_id (3)
	encoder_id (0)	connector_id (0)	connector_id (0)	internal_edid (不配置)	polarity (1000000)
	max_freq (340000)		type (eDP)	type (eDP)	peroid (1)
	max_width (4096)		config_type (driver)	hotplug (connected)	usage (0)
	max_height (4096)		chip_id (ncs8805)	edid_method (I2C)	
			chip_addr (0x3a)		

表 7-4 GPU 配置

设备	GPU	显存
属性/配置	feature (0)	feature (0)
	gpu_freq_level (3)	vram_type (ddr4)
	gpu_freq (500MHz)	vram_bit_width (32 bit)

	num_gpu_shaders (64)	vram_capacity (1 GB)
		vram_freq (2400MHz)

# 龙芯 **LA** 架构硬件设计规范

**V1.0**

## 版本信息

文档信息	文档名	龙芯 LA 架构硬件设计规范
	版本号	V1.0
	创建人	研发部
版本历史		
序号	版本号	更新内容
1	V1.0	初版发布



## 目 录

1 龙芯 3A5000 .....	3
2 龙芯 3A6000 .....	3
3 龙芯 3C5000 .....	3
4 龙芯 3D5000 .....	4
5 龙芯 7A1000 .....	5
6 龙芯 7A2000 .....	6
7 龙芯 2K2000.....	7

## 前 言

本文档作为龙芯 LA 架构芯片的主板硬件设计规范，内含处理器的软件与硬件之间的连接要求、规范。参考该文档做硬件设计时，需要参考对应芯片的硬件设计指导。

如遇内容错误，欢迎批评指正。

## 1 龙芯 3A5000

为规范硬件设计和软件开发，龙芯 3A5000 相关板卡请遵循以下规则：

功能	设计规范	备注
UART	LS3A5000 调试串口固定为 UART0。无特殊情况，请不要随意更换。该串口亦可用作系统下普通串口功能。	
HT	LS3A5000 芯片仅 HT1 可用于与桥片互联。	
I2C	固定 I2C0 作为调压以及读内存 spd。	
看门狗	GPIO 控制：定义 GPIO10 为看门狗使能，GPIO11 为喂狗信号。	

## 2 龙芯 3A6000

为规范硬件设计和软件开发，龙芯 3A6000 相关板卡请遵循以下规则：

功能	设计规范	备注
UART	LS3A6000 调试串口固定为 UART0。无特殊情况，请不要随意更换。该串口亦可用作系统下普通串口功能。	
看门狗	GPIO 控制：定义 GPIO10 为看门狗使能，GPIO11 为喂狗信号。	
I2C	固定 I2C0 作为调压以及读内存 spd。若选用 I2C 作为调压芯片，约定设备地址为 6BH。	
AVS	AVS 调压地址：建议 VDDN 设置成 00h, VDDP 设置成 01h。	

## 3 龙芯 3C5000

为规范硬件设计和软件开发，龙芯 3C5000 相关板卡请遵循以下规则：

功能	设计规范	备注
HT	LS3C5000 芯片仅 HT0 可用于与桥片互联。	
BMC	1、约定 GPIO10 作为固件 post 完成标记；GPIO15 作为 thermtrip 功能。针对使用 2K0500 BMC 的服务器，约定 GPIO14 作为监测 BMC 重启信号。	

	2、约定 UART0 与 BMC 的 UART0 对接（作为 SOL 功能使用）。	
看门狗	GPIO 控制：定义 GPIO6 为看门狗使能，GPIO5 为喂狗信号。	
UART	LS3C5000 调试串口固定为 UART0。无特殊情况，请不要随意更换。该串口亦可用作系统下普通串口功能。	
I2C	固定 I2C1 作为电源调压，I2C0 读内存 spd。	3C5000 I2C 说明见表 1

表 1 3C5000 I2C 说明

3C5000 内存通道	MC0	MC1	MC2	MC3
3C5000L 内存通道	MC2	MC0	MC1	MC3
I2C 通道	I2C0			
I2C 地址	100, 101	000, 001	010, 011	110, 111

## 4 龙芯 3D5000

为规范硬件设计和软件开发，龙芯 3D5000 相关板卡请遵循以下规则：

功能	设计规范	备注
HT	LS3D5000 芯片仅 HT0 可用于与桥片互联。	
BMC	1、约定 GPIO10 作为固件 post 完成标记；GPIO15 作为 thermtrip 功能。针对使用 LS2K0500 BMC 的服务器，约定 GPIO14 作为监测 BMC 重启信号。 2、约定 UART0 与 BMC 的 UART0 对接（作为 SOL 功能使用）。	
看门狗	GPIO 控制：定义 GPIO5 为看门狗使能，GPIO6 为喂狗信号。	
UART	LS3D5000 调试串口固定为 UART0。无特殊情况，请不要随意更换。该串口亦可用作系统下普通串口功能。	
I2C	固定 I2C0 作为电源调压，I2C0 读内存 spd。	3D5000 I2C 说明见表 2

表 2 3D5000 I2C 说明

内存通道	DIE0_M C0	DIE0_MC 1	DIE1_MC 0	DIE1_MC 1	DIE0_MC2	DIE0_M C3	DIE1_MC 2	DIE1_MC 3
I2C通道	DIE0_I2C1				DIE0_I2C2			
I2C地址	000,001	010, 011	100, 101	110, 111	000, 001	010,011	100, 101	110, 111

## 5 龙芯 7A1000

为规范硬件设计和软件开发，龙芯 7A1000 相关板卡请遵循以下规则：

功能	设计规范	备注
RTC	<ol style="list-style-type: none"> <li>默认通用操作系统使用内部 RTC 功能模块。</li> <li>外挂 RTC 需要选用《龙芯外围功能芯片支持列表》中的器件，目前仅支持计时功能，其他功能暂不支持。</li> </ol>	
SPI	<ol style="list-style-type: none"> <li>LS7A1000 桥片的 SPI 总线推荐连接一个 SPI flash，片选固定为 CS0，读写方式必须完全兼容 SST25VF010，该 flash 用于存储桥片集成 GMAC 的 MAC 地址、LS7A1000 的 GPU 显示相关参数信息、主板串号等。</li> <li>使用 SPI 的 Tcm 芯片需要连接到 CS1。</li> </ol>	
GPIO	<ol style="list-style-type: none"> <li>约定使用 GPIO0 作为蜂鸣器控制，高电平有效。</li> <li>使用 SmartFan 功能，约定 PWM0 作为处理器风扇控制，PWM1 作为处理器风扇转速监测。</li> </ol>	
显示	LS7A1000 内置显卡的硬件设计要求、VBIOS 固件的存放位置等，请参见《龙芯 7A1000 内置显卡软硬件设计规范》。	
I2C	如果使用 I2C 触摸板，约定 GPIO50 作为中断输入。	
BMC	连接 BMC 的 PCIE，推荐优先使用主桥的 F0_P1。	
PCIE	<ol style="list-style-type: none"> <li>针对服务器双桥使用场景，副桥只支持 PCIE。</li> <li>针对服务器使用场景，接 PCIe_Riser 卡时，需要把控制器对应的 PCIE 复位控制信号接到相应的槽位上。以 G0/G1/H 控制器为例示意图见图 1。</li> </ol>	

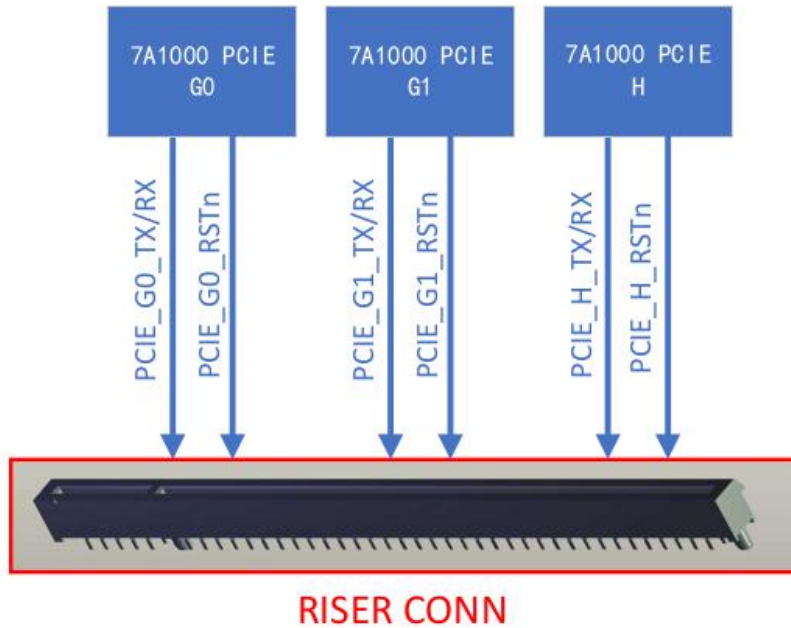


图 1 LS7A1000 PCIe 接 Riser 时连接示意图

## 6 龙芯 7A2000

为规范硬件设计和软件开发，龙芯 7A2000 相关板卡请遵循以下规则：

功能	设计规范	备注
RTC	<ol style="list-style-type: none"> <li>1、默认通用操作系统使用内部 RTC 功能模块。</li> <li>2、外挂 RTC 需要选用《龙芯外围功能芯片支持列表》中的器件，目前仅支持计时功能，其他功能暂不支持。</li> </ol>	
SPI	<ol style="list-style-type: none"> <li>1、LS7A2000 桥片的 SPI 总线推荐连接一个 SPI flash，片选固定为 CS0，读写方式必须完全兼容 SST25VF010，该 flash 用于存储桥片集成 GMAC 的 MAC 地址、LS7A2000 的 GPU 显示相关参数信息、主板串号等。</li> <li>2、使用 SPI 的 Tcm 芯片需要连接到 CS1。</li> </ol>	
GPIO	<ol style="list-style-type: none"> <li>1、约定使用 GPIO0 作为蜂鸣器控制，低电平有效。</li> <li>2、使用 SmartFan 功能，约定 PWM0 作为处理器风扇控制，PWM1 作为处理器风扇转速监测。</li> </ol>	
显示	LS7A2000 内置显卡的硬件设计要求、VBIOS 固件的存放位置等，请参见《龙芯 7A2000 内置显卡软硬件设计规范》。	
I2C	如果使用 I2C 触摸板，约定 GPIO50 作为中断输入。	

BMC	连接 BMC 的 PCIE，推荐使用主桥的 F0_P1。对于 2K0500 BMC，必须使用该端口。	
PCIE	1、针对服务器双桥使用场景，副桥只支持 PCIE, SATA, GNET, USB3。 2、针对服务器使用场景，接 PCIe_Riser 卡时，需要把控制器对应的 PCIE 复位控制信号接到相应的槽位上。以 G0/H 控制器为例示意图见图 2。	

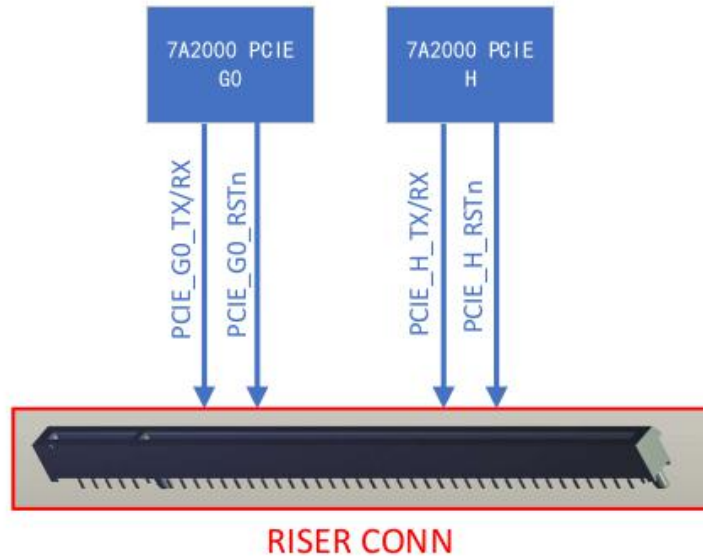


图 2 LS7A2000 PCIE 接 Riser 时连接示意图

## 7 龙芯 2K2000

为规范硬件设计和软件开发，龙芯 2K2000 相关板卡请遵循以下规则：

功能	设计规范	备注
RTC	1. 默认通用操作系统使用内部 RTC 功能模块。 2. 外挂 RTC 需要选用《龙芯外围功能芯片支持列表》中的器件，目前仅支持计时功能，其他功能暂不支持。	
串口	约定 ND_TXD0, ND_RXD0 作为系统调试串口。	
显示	LS2K2000 内置显卡的硬件设计要求、VBIOS 固件的存放位置等，请参见《龙芯 2K2000 内置显卡软硬件设计规范》。	

# 龙芯外围功能芯片支持列表

**V1.2**



## 版本信息

文档信息	文档名:	龙芯外围功能芯片支持列表
	版本号:	V1.2
	创建人:	通用事业部
更新历史		
版本号	更新内容	
V1.0	初版发行	
V1.1	1 内存颗粒增加 CXDQ3BFAM-IJ-A 2 usb 转串口 CH348L 3 增加 spi flash 芯片型号 4 增加 RTC 外置芯片 5 删除 NCS8803	
V1.2	1 新增 RTC 外置芯片	

## 《龙芯外围功能芯片支持列表》

类型	厂商	型号	备注	已适配平台
千兆网卡	瑞昱 (Realtek)	RTL8111G/H		7A1000 全系列、 7A2000 全系列
	网迅	WX1860AL-W		7A1000 全系列、 7A2000 全系列
	Intel	82574	已停产	3A4000-7A1000 产品
		I210/I211		7A1000 全系列
		82580		7A1000 全系列
I350		7A1000 全系列、 7A2000 全系列		
万兆网卡	网迅	WX1820AL		7A1000 全系列、 7A2000 全系列
	Intel	82599		7A1000 全系列、 7A2000 全系列
		X710		7A1000 全系列、 7A2000 全系列
千兆网络 PHY	Marvell	88E1510/88E1512		7A1000 全系列、 7A2000 全系列
		88E1111-BAB1I000		7A1000 全系列
	瑞昱 (Realtek)	RTL8211E		7A1000 全系列
		RTL8211FD		7A2000 全系列
	裕太微	YT8521SC/H		7A1000 全系列、 7A2000 全系列
		YT8531		7A1000 全系列、 7A2000 全系列
		YT8511		7A1000 全系列、 7A2000 全系列
	普维特电子	RPC101	无法关闭 EEE 模式	7A1000 全系列
	Microchip	KSZ9031		7A1000 全系列
PCIE 转 USB3.0	瑞萨	UDP720201	四口	7A1000 全系列、 7A2000 全系列
	钰创	EJ188H	四口	7A1000 全系列、 7A2000 全系列
	祥硕	ASM1042	双口	7A1000 全系列

PCIE 转 USB3.1	祥硕	ASM3142	双口	7A1000 全系列
PCIE 转 SATA3.0	Mavell	9215		7A1000 全系列
	祥硕	ASM1061		
	祥硕	ASM1064		
PCIE Switch	Broadcom	PEX8112		3A/B4000-7A1000 产品
		PEX8619		
		PEX8632		
	祥硕	ASM8024		
PCIE 转 PCI Switch	Broadcom	PEX8112		3A/B4000-7A1000 产品
PCIE ReDriver	万协通信	WSTR62		7A2000 全系列
USB2.0 HUB	江苏沁恒	CH334U		3A5000-7A1000 产品
SuperIO	Winbond	W83527	LPC 接口	7A1000 全系列
		W83795	I2C 接口	7A1000 全系列
	Fintek	F81866AD	LPC 接口	7A1000 全系列
	Nuvoton	NCT6106D	LPC 接口	7A1000 全系列、7A2000 全系列
GPU	景嘉微	JM7201	PCIE x8	7A1000 全系列
		JM9200	PCIE x8	7A1000 全系列
	AMD	R5 340		7A1000 全系列、7A2000 全系列
		Radeon 520		7A1000 全系列、7A2000 全系列
		RX550		7A2000 全系列
DVO 编码器	Chrontel	CH7055A	DVO 转 VGA	7A1000 全系列
	Analog device	ADV7125	DVO 转 VGA	
	深圳晶格微电子	SDA7123	DVO 转 VGA	
	Analog device	ADV7513	DVO 转 HDMI	
	合肥宏晶微电子	MS7210	DVO 转 HDMI	
	Silicon image	Sii9022ACNU	DVO 转 HDMI	
	龙讯半导体	LT8618SXB	DVO 转 HDMI	
	TI	TFP410PAP	DVO 转 DVI	
	振芯科技	GM7510	DVO 转 DVI	
	新港海岸	NCS8805	DVO 转 EDP	
	振芯科技	GM8285C	DVO 转 LVDS	
显示转换芯片	龙讯半导体	LT6711(不支持系统下修改分辨率)	HDMI 转 EDP	7A2000 全系列

		LT9721		
		LT8619	HDMI 转 LVDS	
HDA codec	前海深蕾	CX20632		7A1000 全系列、 7A2000 全系列
	瑞昱 (Realtek)	ALC897		7A1000 全系列、 7A2000 全系列
		ALC269		7A1000 全系列、 7A2000 全系列
		ALC662	已停产	7A1000 全系列
I2S codec	顺芯	ES8388	只能有 1 个输入	7A2000 全系列
Clock gen	IDT	6P41505		7A1000 全系列、 7A2000 全系列
	奥拉	Au5329		7A1000 全系列、 7A2000 全系列
	IDT	5P49V6965A		3A5000-7A2000 产品
	微寰半导体	MSC1421		7A1000 全系列
	IDT	9FG108EFILFT		7A1000 全系列
Clock buff	奥拉	Au5411		7A1000 全系列
	IDT	5PB1104PGGI		3B4000、3B5000、 3C5000L 系列
	ON	NB3V1104CDTR2G		3B4000 系列
UART	江苏沁恒	CH384L	PCIE 转 UART	7A1000 全系列
		CH382		
		CH340/CH341/CH342/C H343/CH344	Usb 转 UART	
		CH9101/CH9102/CH910 3		
		CH9344		
		CH348L		
RapidIO	IDT	TSI721	PCIE 转 Rapid io	7A1000 全系列
DDR4 颗粒	龙芯中科	LS6D4323-I	512M x8	3A4000 全系列 3A5000 全系列
		LS6D4423-I	256M x16	3A4000 全系列 3A5000 全系列
	Hynix	H5ANAG6NCJ	2GB x16	7A2000 全系列
		H5AN4G6NBJR-VKC	512M x16	7A2000 全系列
	Samsung	K4AAG165WA BCTD	2GB x16	7A2000 全系列
		K4A4G165WF BCTD	512M x16	7A2000 全系列

	紫光国芯	SCB12Q4G160AF-07Q		3A4000 全系列
	合肥长鑫	CXDQ3BFAM-CG		3A4000 全系列 7A2000 全系列
		CXDQ3A8AM-CQ		3A4000 全系列 3A5000 全系列
		CXDQ3BFAM-IJ-A		3A5000 全系列
DDR3 颗粒	Samsung	K4B2G1646F-BYMA		7A1000 全系列
		K4B2G1646Q-BCK0		
		K4B1G1646G-BCK0		
		K4B1G1646I-BCNB		
	晶豪科技 (ESMT)	M15T1G1664A		
紫光国芯	HXI15H4G160AF-13K	已停产		
232 串口芯片	南京国博电子	WS3232ECN		7A1000 全系列、 7A2000 全系列
		WS3243FCA		7A1000 全系列、 7A2000 全系列
	英联	UM3232EEUE		3A4000 、 3A5000 、 3C5000L 系列
	芯景科技	AT3232EUE+		3A5000 系列
	Maxim	MAX3243EAI		3A4000 、 3A5000 、 3C5000L 系列
	Maxim	MAX3243CAI		3A4000 、 3A5000 、 3C5000L 系列
SPI Flash	兆易创新	GD25Q16ESIGR	16Mb 3.3V	7A1000 系列、7A2000 系列
		GD25Q64CSIG	64Mb 3.3V	3A4000 系列、3A5000 系列 (需要加电平转换)
		GD25Q127CSIG	128Mb 3.3V	3A4000 系列、3A5000 系列 (需要加电平转换)
	Winbond	W25Q64JVSIG		7A1000 系列、7A2000 系列
		W25Q32JVSIQ	32Mb 3.3V	3A4000 系列、3A5000 系列 (需要加电平转换)
		W25Q64JVSIQ	64Mb 3.3V	3A4000 系列、3A5000 系列 (需要加电平转换)
		W25Q64FVSIQ	64Mb 3.3V	3A4000 系列、3A5000 系列 (需要加电平转换)
	博雅	BY25Q32BSSIG	32Mb 3.3V	3A4000 系列、3A5000 系列 (需要加电平转换)

		BY25Q64ASSIG	64Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		BY25Q128ASSIG	128Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		BY25Q32ALSIG	32Mb 1.8V	3A4000 系列（IO 电需要使用 1.8V）
	武汉新芯	XM25QH32CHIG	32Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		XM25QH64CHIQ	64Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		XM25QH64AHIG	64Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		XM25QH128AHIG	128Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		XM25QH128CHIQ	128Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		XM25QU32CJIG	32Mb 1.8V	3A5000 系列
		XM25QU64AHIG	64Mb 1.8V	3A5000 系列
		XM25QU64CHIQ	64Mb 1.8V	3A5000 系列
		XM25QU128CHIQ	128Mb 1.8V	3A5000 系列
	上海复旦微	FM25Q64	64Mb 3.3V	3A4000 系列
	合肥恒烁	ZB25VQ32ASIG	32Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		ZB25VQ64ASIG	64Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		ZB25VQ128ASIG	128Mb 3.3V	3A4000 系列、3A5000 系列（需要加电平转换）
		ZB25VQ64BSJG	64Mb 3.3V	3A5000 系列（需要加电平转换）
		ZB25VQ128DSJG	128Mb 3.3V	3A5000 系列（需要加电平转换）
		ZB25LQ64ASJG	64Mb 1.8V	3A5000 系列
		ZB25LQ128CSJG	128Mb 1.8V	3A5000 系列
芯天下	XT25Q64DSSIGT	64Mb 1.8V	3A5000 系列	
	XT25Q128DSSIGT	128Mb 1.8V	3A5000 系列	
板载 SSD	安信物联	AXH-US064MTW		7A1000 全系列
	三顿电子	SCUD128GMTWT		7A1000 全系列
	鸿秦科技	HTUSMU064G-WM	64GB	7A1000 全系列
测温芯片	申矽凌	CT75MR		7A1000 全系列

	圣邦微	SGM452		7A1000 全系列
	中科银河	GX21M15U		7A1000 全系列
HT 电平转换 芯片	国微	SM0104E		3A/B5000/3C5000L/3 C5000-7A1000 产品
	润石科技	RS0104YQ		
	圣邦微	SGM4578YTQG20G/TR		
	英联	UM3208UK		
WIFI	瑞昱 (Realtek)	RTL8821ce	PCIE 设备	7A1000 全系列、 7A2000 全系列
		RTL8822ce	PCIE 设备	7A1000 全系列、 7A2000 全系列
		RTL8192cu	USB 设备	7A1000 全系列、 7A2000 全系列
		RTL8188gu	USB 设备	7A1000 全系列、 7A2000 全系列
		RTL8812au	USB 设备	7A1000 全系列、 7A2000 全系列
		RTL8822bu	USB 设备	7A1000 全系列、 7A2000 全系列
RTC 外置芯片	武汉芯景	AT8339AT	I2C 设备	7A1000 全系列、 7A2000 全系列
	Maxim	DS1338Z	I2C 设备	7A1000 全系列、 7A2000 全系列、 2K2000 系列
	上海贝岭	BL5372	I2C 设备	7A1000 全系列、 7A2000 全系列、 2K2000 系列
	大普通信	INS5830B	I2C 设备	7A1000 全系列、 7A2000 全系列、 2K2000 系列
	兴威帆电子	SD2068	I2C 设备	7A1000 全系列、 7A2000 全系列、 2K2000 系列
	成都华微	HWD5372	I2C 设备	7A1000 全系列、 7A2000 全系列、 2K2000 系列