

龙芯中科技术股份有限公司企业标准

龙芯 3 号 CPU 固件与内核接口规范

V3.1

2020-12-18 发布

2020-12-18 实施

龙芯中科技术股份有限公司 批准

文档更新记录	文档名	龙芯 CPU 固件与内核接口规范	
	版本号	V3.0	
	创建人	系统研发部	
	创建日期	2014-04-01	
更新历史			
序号	更新日期	版本号	更新内容
1	2014. 4. 01	V1. 0	发布文档初始版本 V1.0 版。
2	2016. 8. 30	V1. 1	修订版本 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	2017. 4. 18	V1. 2	修订版本 V1. 2 1、添加传参新成员 cpuname 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（efi_reset_system_t）； 4、添加成员 DoSuspend；
4	2017. 5. 27	V2. 0	修订版本 V2. 0 1、重新修订附录 B bootparam.h 文件； 2、修正附录 A 中的几处错误；
5	2018. 3. 27	V2. 1	修订版本 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（boardinfo、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	2018. 11. 20	V2. 2	修订版本 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 board_devices.name 受同样约束； 3、附录 A.2，删除现在接口中已经废弃的 screen_info 和 sys_desc_table； 4、附录 A.3，删除现在接口中已经废弃的 systab 和 UEFI runtime 服务相关的接口； 5、附录 A.4，规范了使用龙芯集显情况 vbios 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 of_dtb_addr 成员； 8、附录 A.12，对固件传参接口 cpuname 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	2019.10.10	V2.3	修订版本 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； 3、5.1.2 节，修改中断处理方式，参考各平台固件开发规范； 4、5.2 节，约束 LPC 接口使用方式； 5、第 7 节，修改 SMBIOS 约定必须传递的信息以及实现参考； 6、附录 A.4，更名 smbios_tables 为 sysinfo_tables； 7、附录 A.7，修改 vers 成员的定义、根据实际使用情况修改其他成员定义； 8、附录 B，更新 bootparam.h 文件； 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； 10、修改了一些语法错误和排版问题。
8	2020.12.20	V3.0	修订版本 V3.0 1、文档名称及正文中删除“开发系统” 2、2 节，增加了约定章节和部分术语 3、4.2 节，修改了地址空间的解释 4、5.1.1 节，修改了固件内中断的分工 5、6 节，改变了固件与内核的传参接口 6、7 节，具体化了 type2 中 product 的格式约束 7、8 节，新增了固件对 ACPI 规范的实现约束 8、附录 A，更新为最新固件内核传参接口的的数据结构 9、附录 B，增加 LINUX 操作系统键值表 10、修改了一些语句描述和排版问题。
9	2021.05.27	V3.1	修订版本 V3.1 1、8.5 节，FADT 的 flags 中新增支持 PCI_EXP_WAK 及 RESET_REG_SUP 标志使用描述。

			<div>2、8.3 节、中断模型更新</div> <div>3、增加 8.8 节 MCFG 支持</div> <div>4、8.6.1 节，新增_SEG 对象支持双桥</div> <div>5、6.3.1 节，3.0 接口 bpi 结构增加 64 位 flags</div> <div>6、8.6.9 节，热键驱动增加 VCBL 的约束</div> <div>7、8.6.12 节，GPIO 多中断支持</div> <div>8、8.6.18 节，增加 ACPI 对 SE 设备的支持</div> <div>9、8.6.19 节，增加 ACPI 对温度传感器的支持</div> <div>10、8.9 节，增加 SLIT 表的支持</div> <div>11、8.10 节，增加 SPCR 表的支持</div>
--	--	--	--

目 录

前 言	1
1 范围	2
2 术语与约定	2
2.1 术语	2
2.2 约定	2
3 架构关系	2
4 地址空间规范	3
4.1 地址空间	3
4.2 DMA 地址映射规范	3
5 中断配置规范	3
5.1 配置方法	3
6 固件与内核接口传参规范	3
6.1 参数构成	3
6.2 BootParamsInterface 数据结构的约定	3
6.3 传参的实现	4
6.3.1 BootParamsInterface 定义	4
6.3.2 扩展参数链表	4
6.3.3 扩展参数链表的实现约定	5
6.3.4 MEM_MAP	5
6.3.5 VBIOS	7
6.3.6 ScreenInfo	7
7 SMBIOS 规范的实现约定	8
8 ACPI 规范的实现约定	10
8.1 RSDP (Root System Description Pointer)	11
8.2 XSDT (Extended System Description Table)	12
8.3 MADT (Multiple APIC Description Table)	12
8.4 SRAT (System Resource Affinity Table)	15
8.5 FADT (Fixed ACPI Description Table)	17
8.6 DSDT (Differentiated System Description Table)	19
8.6.1 PCI 总线枚举	19
8.6.2 PCI 中断路由	21
8.6.3 设备电源管理	21
8.6.4 USB 设备配置	21
8.6.5 电池配置	21
8.6.6 电源适配器配置	22
8.6.7 处理器配置和控制	22
8.6.8 系统休眠唤醒	22
8.6.9 热键配置	22

8.6.10 热区管理	24
8.6.11 串口配置	24
8.6.12 GPIO 配置	25
8.6.13 I2C 配置	26
8.6.14 GPIO 模拟 I2C 配置	26
8.6.15 RTC 配置	26
8.6.16 PWM 配置	27
8.6.17 S3 休眠地址	27
8.6.18 SE 设备	27
8.6.19 温度传感器配置	27
8.7 FACS (Firmware ACPI Control Structure)	28
8.8 MCFG (PCI Express Memory-mapped Configuration Space base address description table)	28
8.9 SLIT (System Locality Distance Information Table)	28
8.10 SPCR (Serial Port Console Redirection Table)	29
9 总结	29
附录 A 龙芯 CPU 传参数据结构	30
附录 B LINUX 操作系统键值表	31

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。

本规范涉及到固件与内核之间的传参、中断分配及地址空间划分等方面，主要介绍龙芯板卡固件与内核之间的接口定义，地址空间分配，中断在内核和固件之间的划分、中断号的分配以及 SMBIOS 中需要实现的类型约定。接口定义涉及到固件与内核间信息传递的数据结构定义，固件使用何种方式传递这些数据结构及内核如何解析等方面。

1 范围

本规范规定了龙芯 CPU 的地址空间、固件与内核接口传参实现、ACPI 及 SMBIOS 实现约定的要求。本规范适用于龙芯 LoongArch 架构的 3 号系列 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。针对不同的处理器、芯片组，请参阅对应处理器、芯片组的固件开发规范。

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。

PMON：龙芯平台使用的一种兼有 BIOS 和 boot loader 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：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。

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：

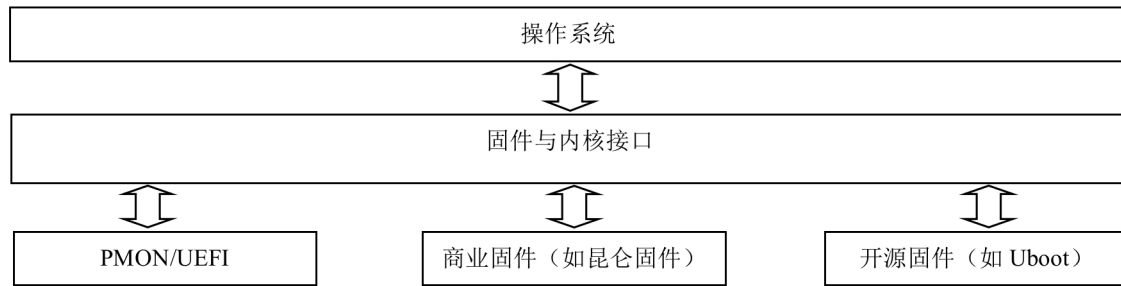


图 3-1 接口与内核和固件之间的关系

4 地址空间规范

4.1 地址空间

龙芯 3 号 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 固件与内核接口传参规范

6.1 参数构成

龙芯固件传递给内核的参数包括内核命令行参数个数、内核命令行参数地址以及 BootParamsInterface 数据结构地址，分别通过 a0、a1 以及 a2 三个寄存器传递给内核。其中，BootParamsInterface 数据结构包括 SMBIOS 表、ACPI 表、扩展参数链表三部分，其地址为 64 位虚拟地址；内核命令行参数个数为 32 位有符号数据；内核命令行参数地址在 LoongArch 架构的龙芯平台上为 64 位虚拟地址。

6.2 BootParamsInterface 数据结构的约定

1. 采用 1 字节对齐方式。
2. 用于存储该数据结构的内存类型为 RuntimeServicesData（详情可参考 UEFI 规范）。

6.3 传参的实现

6.3.1 BootParamsInterface 定义

```
struct BootParamsInterface {
    UINT64 Signature;
    EFI_SYSTEM_TABLE *SystemTable;
    EXT_LIST *ExtList;
    UINT64 Flags;
};
```

BootParamsInterface 由 4 部分组成：标签（Signature）、系统表（SystemTable）、扩展参数链表（ExtList）和标志（Flags）。

表 6-1 BootParamsInterface 结构

域	大小	描述
Signature	u64	签名标识及版本号。形式为‘BPIXYYY’的 ASCII 码字符串。前 3 字节固定为字符‘BPI’，‘XX’为大版本号，‘YYY’为小版本号。‘X’，‘Y’的取值范围为 ASCII 码的‘0’-‘9’。当前版本为“BPI01001”。
SystemTable	u64	指向 UEFI 构建的系统表
ExtList	u64	指向扩展参数链表。扩展参数链表作为标准工业规范（ACPI、SMBIOS）之外的一个补充，包含了平台必须的一些参数。详情参考章节 6.3.2
Flags	u64	标志。 bit 0: 0 表示固件支持 UEFI 规范； 1 表示固件不支持 UEFI 规范。 bit 1: 0 表示基于非龙芯 SOC 处理器设计，主板集成了芯片组； 1 表示基于龙芯 SOC 处理器设计，主板无芯片组。 [63:2]位：为 0。

注：SystemTable 的成员 ConfigurationTable 包含了 SMBIOS 表、ACPI 表的入口地址，详细描述参考章节 7 SMBIOS 的实现约定和章节 8 ACPI 的实现约定。

6.3.2 扩展参数链表

扩展参数链表为一个单向链表，每个链表结点描述了一个平台相关配置，链表整体示意图 6-3 如下：

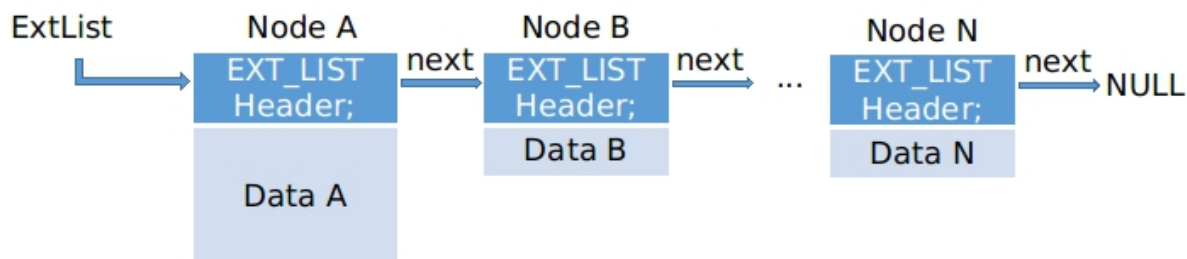


图 6-3 扩展参数链表示意图

每个结点都由 Header 段和 Data 段构成，其中 Header 用于组成扩展链表，Data 是结点的私有数据。用户可根据实际需求创建结点，添加到链表然后传递给内核。

EXT_LIST 定义如下：

```
typedef struct _extension_list_hdr{
    UINT64  Signature;
    UINT32  Length;
    UINT8   Revision;
    UINT8   CheckSum;
    struct _extension_list_hdr *next;
}EXT_LIST;
```

表 6-2 EXT_LIST 结构

域	大小	描述
Signature	u64	结点的唯一标识符，由 ASCII 码字符串表示，不满 8 字节部分用 0 补充
Length	u32	结点的长度，单位字节（Byte），从结点偏移 0 开始到结点结束，用于记录整个结点大小
Revision	u8	结点结构的修订版本号。版本号的升级要保证向前兼容
CheckSum	u8	校验和，整个结点包括 CheckSum 字段加起来的和为 0
next	u64	下一结点的 64 位虚拟地址

6.3.3 扩展参数链表的实现约定

结点可根据实际需求进行创建，本规范定义了必须实现的扩展结点，结点签名及描述见表 6-3。

表 6-3 扩展参数链表结点签名

签名（Signature）	描述	参考
‘MEM’	系统的内存映射表	章节 6.3.4
‘VBIOS’	VBIOS 的地址	章节 6.3.5

6.3.4 MEM_MAP

MEM_MAP 描述了系统的内存布局和内存属性，结构定义如下，说明见表 6-4

```
typedef struct {
    EXT_LIST Header;
    UINT8   MapCount;
```

```

struct  MemMap {
    UINT32 MemType;
    UINT64 MemStart;
    UINT64 MemSize;
}Map[MAX_MEM_MAP];
} MEM_MAP;

```

表 6-4MEM_MAP 结构

域	大小	描述
Header		
Signature	u64	‘MEM’
Revision	u8	0
MapCount	u8	Map 数组中有效成员个数
MemMap		
MemType	u32	内存类型，参考表 6-5
MemStart	u64	内存的 64 位起始地址，其中 [47:44] 位表示内存所在的 CPU 节点号
MemSize	u64	内存的大小，单位字节（Byte）

注意：

- 1 龙芯地址空间是统一编址的，传递的内存映射表应该满足：
 - a. usable 区域和 reserved 的区域不能够重合；
 - b. 假设所有可用内存（SYSTEM_RAM）区域的最小起始地址是 USABLE_START, 最大结束地址是 USABLE_END，要求：任何一项保留内存（MEM_RESERVED）区域的起始地址大于 USABLE_START，结束地址小于 USABLE_END。示意如图 6-4：

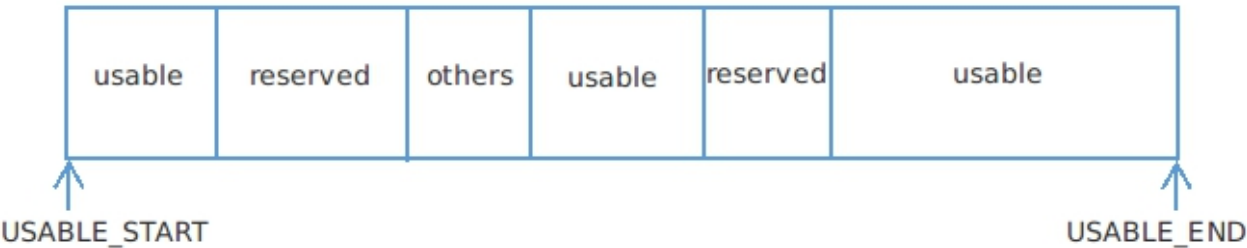


图 6-4 内存区域示意图

- 2 内存大小的单位是字节，与接口规范 V2.3 定义不同。

表 6-5 内存类型

内存类型	值	描述
SYSTEM_RAM	1	系统内存
MEM_RESERVED	2	系统保留内存
ACPI_TABLE	3	ACPI 表存储内存
ACPI_NVS	4	ACPI 预留内存

6.3.5 VBIOS

该表描述了 VBIOS 的存储地址，见表 6-6 解释。

```
typedef struct {
    EXT_LIST Header;
    UINT64 VbiosAddr;
}VBIOS;
```

表 6-6 VBIOS 结构

域	大小	描述
Header		
Signature	u64	‘VBIOS’
Revision	u8	0
VbiosAddr	u64	Vbios 的入口地址，64 位虚拟地址

6.3.6 ScreenInfo

该表仅适用于 UEFI BIOS，描述了显卡的 framebuffer 信息，包括显存基地址和显存分辨率等，主要应用于虚拟化场景。对于多个显卡，仅描述主显卡的 framebuffer 信息。

该表描述了 SINFO 的存储地址，见表 6-7 解释。

```
typedef struct {
    EXT_LIST Header;
    UINT64 SinfoAddr;
}SINFO;
```

表 6-7 SINFO 结构

域	大小	描述
Header		
Signature	u64	‘SINFO’
Revision	u8	0
SinfoAddr	u64	指向 screen_info 结构体类型的指针，和 Linux 内核 struct screen_info 保持一致

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 6-8：

```
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 */
};
```

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 系统表的配置表中，内核在配置表中查找 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：

- 1) BIOS 信息 (Type 0)
- 2) 系统信息 (Type 1)
- 3) 产品信息 (Type 2)
- 4) 系统外围或底架 (Type 3)
- 5) 处理器信息 (Type 4)
- 6) 高速缓存信息 (Type 7)
- 7) 系统插槽 (Type 9)
- 8) 物理存储阵列 (Type 16)
- 9) 存储设备 (Type 17)
- 10) 存储阵列映射信息 (Type 19)
- 11) 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，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。其中第六个字节的的信息，里面包含了这个主机的类型。见表 7-9。

表 7-9 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，对应接口规范 V2.3efi_cpuinfo_loongson.cpu_clock_freq；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 8-10：

表 8-10 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

8.1 RSDP (Root System Description Pointer)

RSDP 是整个 ACPI 表的第一个表，与其他表的关系可以用下图表示。

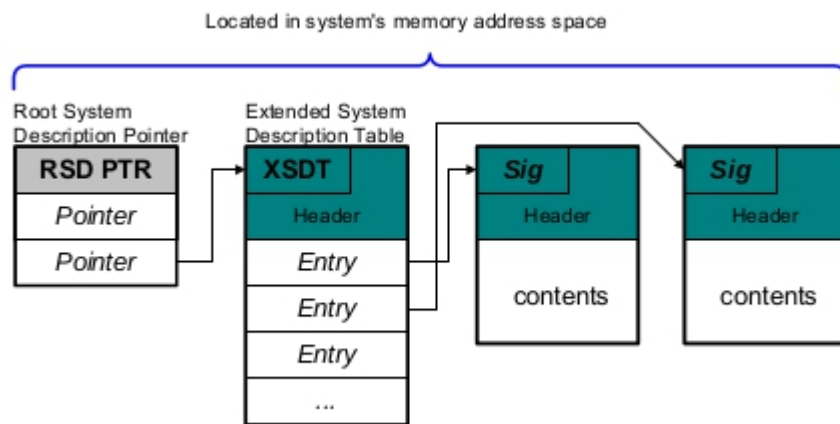


图 8-5 RSDP 示意图

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 的地址存储在 UEFI 系统表（SystemTables）的配置表（ConfigurationTables）中，内核通过在配置表中查找 ACPI2.0 GUID（{8868e871-e4f1-11d3-bc22-0080c73c8881}）来获取 RSDP 的地址。

RSDP 填写规范见表 8-11：

表 8-11 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	2.
RsdtdAddress	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 (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 8-12：

表 8-12 XSDT 表约束

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

8.3 MADT (Multiple APIC Description Table)

本规范使用 MADT 表描述系统多核信息。

表 8-17，表列出了 MADT 表部分约束条件，未列出的部分参考 ACPI 规范。

系统中的每个处理器核都需要在 MADT 中有一个 Local APIC 记录，用于操作系统多核启动及处理器中断控制器信息，填写要求见表 8-13，IO APIC 结构用于描述芯片组中断控制器信息，填写要求见表 8-16。

表 8-13 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	处理器中断控制器寄存器基地址
Flags	4	40	0
Interrupt		44	中断控制器结构列表。本规范仅实现 Local APIC 结构，参考 8-14

Controller Structure[n]			的实现要求
----------------------------	--	--	-------

MADT 表的 CORE PIC 结构参考如下配置：

表 8-14 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 8-15

表 8-15 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 结构参考如下配置：

表 8-16 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 表的 HT PIC 结构参考如下配置：

表 4-6 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 表的 EIO PIC 结构参考如下配置：

表 8-17 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 表的 MSI PIC 结构参考如下配置：

表 8-18 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 表的 BIO PIC 结构参考如下配置：

表 8-19 BIO PIC 结构

域	大小 (字节)	偏移 (字节)	描述
Type	1	0	BI0 PIC 结构类型
Length	1	1	BI0 PIC 结构字节长度
Version	1	2	版本号
Base Address	8	3	BI0 PIC 寄存器的基地址
Size	2	11	BI0 PIC 寄存器空间的大小
Hardware ID	2	13	BI0 PIC 的硬件 ID，即 BI0 PIC 所在芯片组连接的处理器节点的节点号
GSI base	2	15	BI0 PIC 中断开始的 GSI 号，对于每个中断的 GSI， $GSI = GSI\ base + BI0\ PIC\ 的中断向量$

MADT 表的 LPC PIC 结构参考如下配置：

表 8-20 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 BI0 PIC 的 BI0 PIC 向量号

8.4SRAT (System Resource Affinity Table)

此表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 8-21。

龙芯平台需要实现两种设备的亲和表：

1 “Processor Local APIC/SAPIC Affinity Structure”，每个处理器核对应一个结构，见表 8-22；

2 “Memory Affinity Structure”，每个具有内存的处理器节点的低端内存和高端内存分别对应一个结构，见表 8-23。

表 8-21 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 8-22 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 8-23 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 8-24
Reserved	8	32	保留

表 8-24 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	是否支持内容热插拔
NonVolatile	1	2	是否为非易失内存
Reserved	29	3	0

8.5 FADT (Fixed ACPI Description Table)

此表为操作系统提供了 Fixed 硬件 ACPI 描述信息。

表 8-25 列出了 FADT 表部分约束条件，未列出的部分请参考 ACPI 规范。

表 8-25 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	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 8-26
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 8-22
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 8-22
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 8-22
X_GPE0_BLK	12	220	GPE0 block 的 64 位地址，见表 8-26

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 8-26 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 (Differentiated System Description Table)

本表用来描述主板设计差异化信息。

本章节规定了龙芯平台对 DSDT 配置的支持范围，此范围之外的配置未验证，不保证其正确性。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。

- _CRS (Current Resource Settings)

(1) _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 8-27：

表 8-27 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 WordIO() 宏声明 PCI IO 地址范围，各个参数规定如表 8-28:

表 8-28 WordIO() 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 DWordMemory() 宏声明 PCI MEMORY 地址范围，各个参数规定如表 8-29:

表 8-29 DWordMemory 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。

● _PR0 (Power State 0)

- (1) 必须提供这个控制方法。

● _PRW (Power Resources for Wake)

- (1) 仅支持 EventInfo 和 DeepestSleepState 参数。

8.6.3.2 GMAC

支持的对象及方法：_PR0、_PRW。

● _PR0 (Power State 0)

- (1) 必须提供这个控制方法。

● _PRW (Power Resources for Wake)

- (1) 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。

8.6.6 电源适配器配置

支持的对象及方法：_PSR 。

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。

● _PCT(Performance Control)

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

● _PSS (Performance Supported States)

(1) PSS 中的 Power 、Bus master latency、Status 为任意值。

(2) Latency: 大于 20000

(3) 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

龙芯平台提供了统一的热键驱动，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 L00N0000。

(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 8-30：

表 8-30 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 B

(3) 热键事件产生时，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 8-31：

表 8-31 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

（4）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 B。

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

（5）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，则由集显背光设备控制背光。

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

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。

- _HID

(1) 龙芯平台为 PNP0501。

- _UID

(1) 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。

- _CRS

(1) _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 8-25。中断号资源使用 Interrupt() 宏声明，各个参数的规定如表 8-32：

表 8-32 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

● _DSD

(1) _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 8-33：

表 8-33 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，只支持芯片组集成的 GPIO。

● _HID

(1) 芯片组集成的普通 GPIO 为 L00N0002，处理器集成的 GPIO 为 L00N0007。

● _CRS

(1) _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 8-25，中断号资源使用 Interrupt() 宏声明，各个参数的规定如表 8-28。

● _DSD

(1) _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 8-34：

表 8-34 支持属性如表

属性名	说明
conf_offset	寄存器起始地址相对基地址偏移
out_offset	输出寄存器相对基地址偏移
in_offset	输入寄存器相对基地址偏移
int_ctrl_offset	中断控制寄存器相对基地址偏移
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
gsi_idx_map	各 GPIO 与 InterruptList 对象中的中断号对应关系

8.6.13 I2C 配置

支持的对象及方法：_HID，_UID，_CRS。

- _HID
 - (1) 龙芯平台为 L00N0004。
- _UID
 - (1) 必须为表示 I2C 总线号的任意正整数。
- _CRS
 - (1) _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 8-25。

8.6.14 GPIO 模拟 I2C 配置

支持的对象及方法：_HID，_UID，_DSD，_CSR。

- _HID
 - (1) 龙芯平台为 L00N0005。
 - _UID
 - 必须为表示 I2C 总线号的任意正整数。
 - _DSD
 - (1) _DSD 对象仅支持如下属性：
UUID：值为“daffd814-6eba-4d8c-8a91-bc9bbf4aa301”。
- 支持属性如表 8-35：

表 8-35 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。

- _HID
 - (1) 龙芯平台为 L00N0001。
- _CRS
 - (1) _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 8-25，中断号资源使用 Interrupt() 宏声明，各个参数的规定如表 8-28。

8.6.16 PWM 配置

支持的对象及方法：_HID，_UID，_CRS。

- _HID

(1) 龙芯平台为 LOON0006。

- _CRS

(1) _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 8-25，中断号资源使用 Interrupt() 宏声明，各个参数的规定如表 8-28。

8.6.17 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：

(3) 使用 Name 操作符定义的整数对象，对象名称：SADR。

(4) 该对象需要声明在 _SB 下。

8.6.18 SE 设备

支持的对象及方法：_HID、_CRS。

- _HID

(1) 龙芯平台为 LOON0003。

- _CRS

(1) _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 8-28。

8.6.19 温度传感器配置

支持的对象及方法：_HID，_DSD，_CRS。

- _HID

(1) 龙芯平台为 LOON0008。

- _DSD

(1) _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 8-36：

表 8-36 支持属性如表

属性名	说明
adjust-frequency	温度传感器检测的温度变化是否可以影响处理器频率的调节 0：温度不影响处理器频率 1：温度影响处理器频率

- _CRS

(1) _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 8-25，中断号资源使用 Interrupt() 宏声明，各个参数的规定如表 8-28。

8.7 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 8-37，其他字段保留为 0。

表 8-37 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 (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。此表在双 7A 环境必须实现，表头部分约定如表 8-38。PCIE 配置信息部分约定如表 8-39。

表 8-38 MCFG 约束

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

表 8-39 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 (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 3-40。

表 8-40 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 $x = y$, 相同结点相对距离 当 $x \neq y$, 不同节结点相对距离 xy 取值范围 : $\leq \text{Number of System Localities} - 1$

8.10 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 8-41。

表 8-41 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

9 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、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 龙芯 CPU 传参数据结构

```
#define SYSTEM_RAM          1
#define MEM_RESERVE         2
#define ACPI_TABLE          3
#define ACPI_NVS             4
#define MAX_MEMORY_TYPE     5

#define MAX_MEM_MAP         128

#pragma pack(1)
typedef struct _extension_list_hdr{
    UINT64  Signature;
    UINT32  Length;
    UINT8   Revision;
    UINT8   CheckSum;
    struct _extension_list_hdr *next;
}EXT_LIST;

struct BootParamsInterface {
    UINT64      Signature;
    EFI_SYSTEM_TABLE *SystemTable;
    EXT_LIST     *ExtList;
};

typedef struct {
    EXT_LIST Header;
    UINT8    MapCount;
    struct MemMap {
        UINT32 MemType;
        UINT64 MemStart;
        UINT64 MemSize;
    }Map[MAX_MEM_MAP];
} MEM_MAP;

typedef struct {
    EXT_LIST Header;
    UINT64  VbiosAddr;
}VBIOS;

typedef struct {
    EXT_LIST Header;
    UINT64  SinfoAddr;
```

```
}SINFO;  
#pragma pack()
```

附录 B 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
--------	------