

PHYTIUM 飞腾

Processor Base Firmware 接口规范

(V3.0)

2020 年 12 月

天津飞腾信息技术有限公司

www.phytium.com.cn

版权所有© 天津飞腾信息技术有限公司 2020

此文档用于指导用户的相关应用和开发工作。天津飞腾信息技术有限公司对此文档内容拥有版权，并受法律保护

免责声明©天津飞腾信息技术有限公司对本文档内容有解释权，且保留持续修改的权利

当前版本

文件标识	P-U-SW-PBF
当前版本	3.0
作者	
完成日期	2020.12.1

更新记录

本文档经过如下修订。

发布日期	版本号	私密性	改变备注
2016.06	1.0.0	公开	Phytium Base Package 程序接口规范 1.0
2018.06.1	2.0.0	公开	将 PBP 改为 PBF，修改版本号相关调用接口
2018.06.21	2.0.1	公开	修改处理器、内存接口，增加 PCI-E 控制器接口
2019.03.20	2.1	公开	版本号改为 2 段式；功能号使用 SiP 段；规范调用错误码；扩展处理器、PCI-E 等信息
2019.09.03	2.2	公开	PSS 为 (Phytium System Service) 缩写，修改 PCI_HOST_BRIDGE 的字段描述，增加 GET_UFW_BASE、GET_CFGTBL_INFO 服务
2019.10.11	2.3	公开	更换文档模板；增加固件使用飞腾 logo 规则
2020.04.14	2.4	公开	增加 GET_RESET_MODE、SET_S3_TIMEOUT、CPUFREQ_INFO、CPUFREQ_RW 服务

2020.08.17	2.5	公开	修改图 5-2 中飞腾 logo 位置；明确 CPUFREQ_RW 的 cpu_id 含义；GET_UFW_BASE 改名为 GET_SFW_BASE；修改 CPU_VERSION 服务的调用接口，增加处理器型号和 ID 信息描述；增加 SECURE_REG_RW、PFDI_REGISTER、PFDI_DONE 和 PBF_VERSION_DESCR。
2020.12.17	3.0	公开	PBF 的“P”更新为 Processor；第 3 章增加固件层次描述，更新图 3.1；新增 3.1“固件执行流程”；更新表 4-4；增加 GET_PARAMETER_VERSION、GET_RST_SOURCE、PLL_INIT、PCIE_INIT、DDR_INIT、RELOACTE、DEBUG_INIT、SECURITY_CFG 等服务。

目 录

更新记录	1
1 范畴	1
2 定义与缩写	2
2.1 定义.....	2
2.2 缩写.....	2
3 基于 PBF 的系统软件栈.....	3
3.1 固件执行流程.....	4
4 飞腾系统服务调用接口	5
4.1 飞腾系统服务调用约定.....	5
4.2 飞腾系统服务.....	8
4.2.1 PSSI_VERSION.....	8
4.2.2 PBF_VERSION.....	8
4.2.3 PBF_VERSION_DESCR.....	9
4.2.4 CPU_VERSION.....	10
4.2.5 CPU_CORE_MAPS.....	11
4.2.6 CPU_CORE_CONF	12
4.2.7 MEM_REGIONS	13
4.2.8 MCU_DIMMS	14
4.2.9 PCI_CONTROLLER	15
4.2.10 PCI_HOST_BRIDGE.....	17
4.2.11 GET_SFW_BASE	18
4.2.12 GET_CFGTBL_INFO	19
4.2.13 GET_RESET_MODE.....	19
4.2.14 SET_SUSPEND_TIMEOUT.....	20

4.2.15	CPUFREQ_INFO	20
4.2.16	CPUFREQ_RW	21
4.2.17	SECURE_REG_RW	23
4.2.18	PFDI_RIGISTER.....	24
4.2.19	PFDI_DONE.....	26
4.2.20	GET_PARAMETER_VERSION.....	26
4.2.21	GET_RST_SOURCE.....	27
4.2.22	PLL_INIT	27
4.2.23	PCIE_INIT	28
4.2.24	DDR_INIT_SERVICES.....	30
4.2.25	RELOCATE	34
4.2.26	DEBUG_INIT	35
4.2.27	SECURITY_CFG	35
5	飞腾 LOGO 的固件使用规则	36
	参考文献.....	37

图目录

图 3-1 PBF 软件栈.....	3
图 3-2 基于 PBF 的固件执行流程.....	4
图 4-1 PFDI 机制.....	24
图 5-1 飞腾 LOGO.....	36
图 5-2 固件使用飞腾 LOGO 示意图.....	36

表目录

表 4-1 飞腾系统服务总表.....	5
表 4-2 PBF 服务调用错误码.....	7
表 4-3 PBF 版本描述表.....	9
表 4-4 飞腾处理器型号描述表.....	10
表 4-5 处理器核存在描述表.....	12
表 4-6 处理器配置描述表.....	13
表 4-7 内存地址空间描述表.....	14
表 4-8 内存地址块.....	14
表 4-9 MCU DIMM 信息描述表.....	15
表 4-10 MCU DIMM 描述块.....	15
表 4-11 PCI 控制器描述表.....	16
表 4-12 控制器描述块.....	16
表 4-13 PCI-E Host Bridge 描述结构.....	17
表 4-14 Host Bridge 描述块.....	18
表 4-15 处理器核频率信息表.....	21
表 4-16 PFDI 服务表.....	25
表 4-17 PFDI 服务函数表.....	25

表 4-18	PLL 服务参数表	28
表 4-19	PCIE 服务参数表	28
表 4-20	DDR 初始化服务表	30
表 4-21	DDR 服务参数表	31
表 4-22	PBF 基础服务参数表	34

PHYTIUM

1 范畴

此文档适用于基于飞腾处理器的 Processor Base Firmware 固件。

PHYTIUM

2 定义与缩写

2.1 定义

2.2 缩写

PBF	Processor Base Firmware
PBR	Phytium Boot ROM
PFDI	Phytium Firmware Dispatch Interface
PSCI	Power State Coordination Interface
PSSI	Phytium System Service Interface
SMCCC	SMC Calling Convention
SCMI	System Control and Management Interface
SFW	System Firmware
SiP	Silicon Partner, i.e. the silicon manufacturer
MCU	Memory Controller Unit

3 基于 PBF 的系统软件栈

基于 Processor Base Firmware 的系统软件栈如图 3-1 所示。其中，固件分为三层：飞腾芯片内置可信根（Phytium Boot ROM, PBR）、飞腾处理器基础固件（Processor Base Firmware, PBF）和系统固件（System Firmware, SFW）。

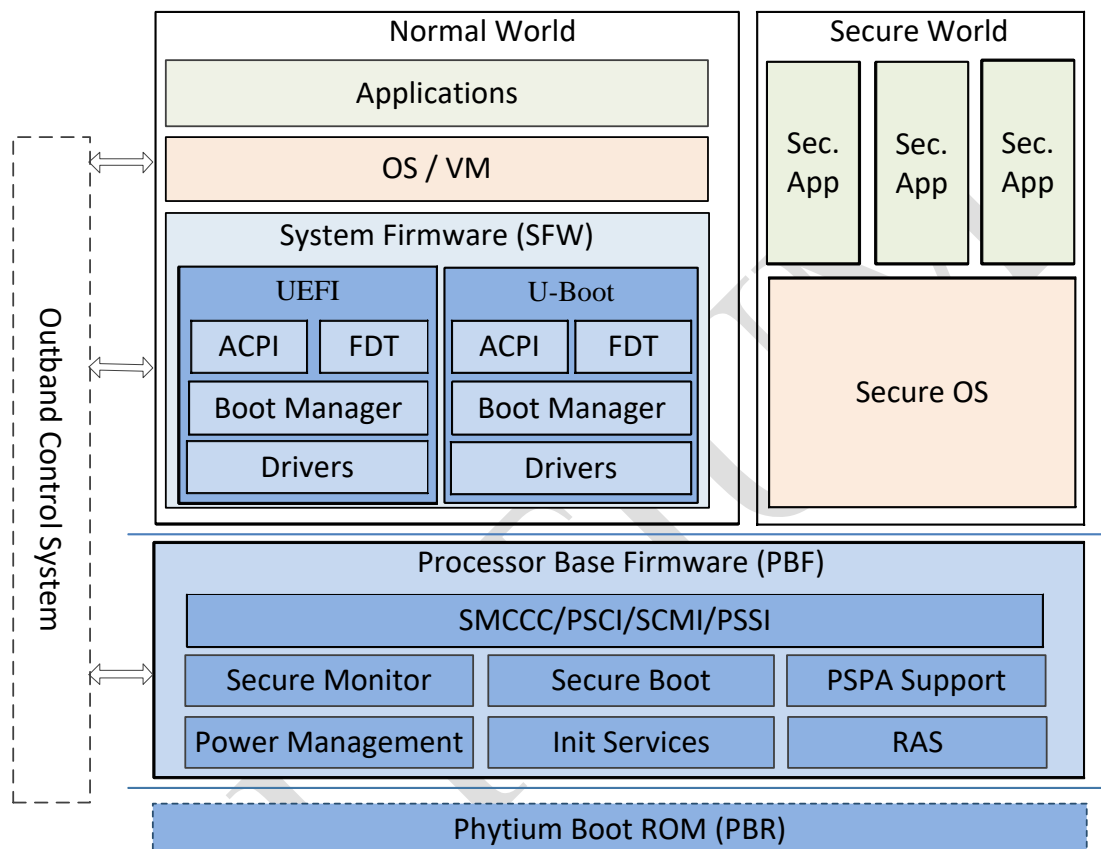


图 3-1 PBF 软件栈

PBR 是飞腾处理器芯片内置的可信启动根，负责对 PBF 进行验签。并非所有飞腾处理器版本都内置 PBR，只有安全版芯片才支持 PBR。PBF 是飞腾处理器基础固件，负责处理器芯片的基本初始化，并提供相关的服务。PBF 还负责加载运行在安全态（Secure World）下的 Secure OS。根据应用领域的不同，飞腾平台的系统固件 SFW 有两种技术路线：面向桌面、服务器等领域的 UEFI 和面向嵌入式领域的 U-Boot。PBF、SFW、OS 等都会与带外控制系统（Outband Control System），比如 EC、BMC 等进行通信。

3.1 固件执行流程

处理器加电后，首先运行位于 Flash 中的 PBF（安全版芯片加电后首先运行处理器芯片内 PBR，然后再运行位于 Flash 中的 PBF）。PBF 运行在 EL3，完成最基本的处理器芯片初始化后，跳转到位于 Flash 中的 System Firmware。

System Firmware 运行在 EL2，根据系统需要，依次调用 PBF 提供的各模块初始化服务，比如 PLL、SOC 内存控制器、DDR、PCIe 控制器、SoC 模块等，完成相应模块的初始化。PBF 还提供一些查询复位，比如，查询系统复位的原因（Reset Source）。

PBF 第一次进入 System Firmware 时，并未初始化内存，PBF 没有内存可用，功能受限。为此，System Firmware 基于 PBF 的内存初始化服务（Mem Ctrl、DDR 等），完成内存初始化后，必须显式调用 PBF 提供的重定位服务（Relocation），由 PBF 重定位服务将 PBF 自身加载到内存中，并完成 PBF 服务的内存重定位。System Firmware 完成相关的系统初始化后，加载操作系统。

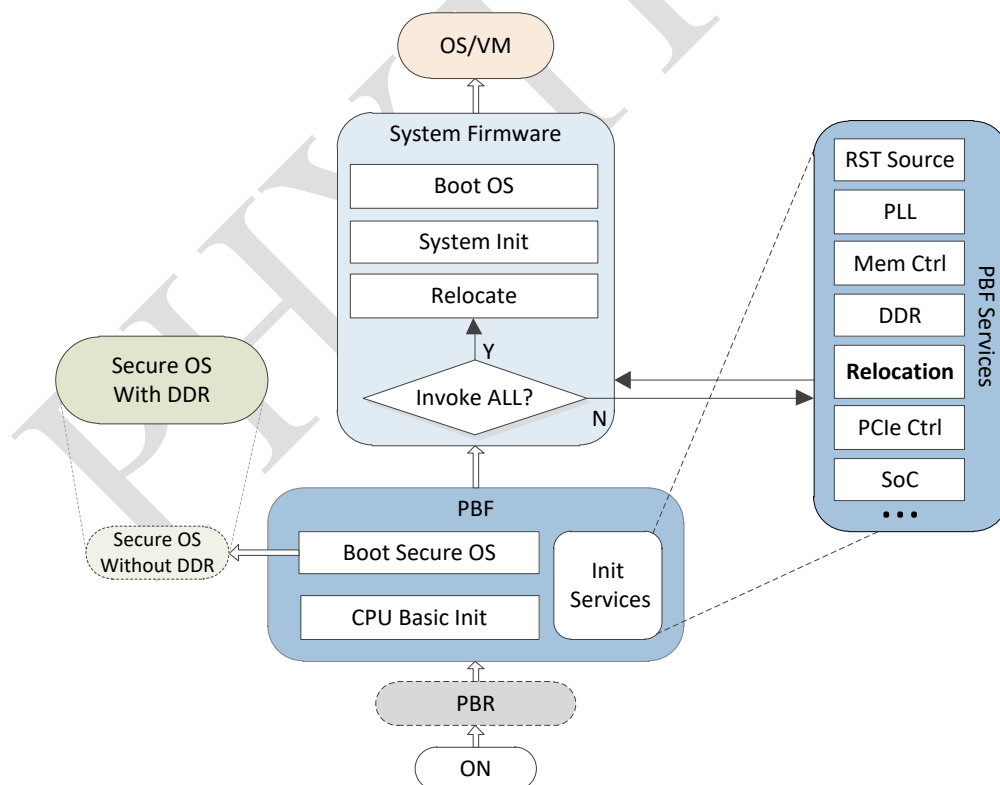


图 3-2 基于 PBF 的固件执行流程

在需要加载运行 Secure OS 的环境中，PBF 重定位服务还负责 Secure OS 的内存重定位。

4 飞腾系统服务调用接口

PBF 是飞腾处理器基础固件，负责飞腾处理器实现相关的硬件初始化，为了简化第三方软件开发，特写此规范供参考。

PBF 向上层软件提供 PSCI 接口服务和获取平台信息的飞腾系统服务，本文档仅描述由 Phytium 实现的飞腾系统服务，PSCI 接口请参考《Power State Coordination Interface》文档^[2]。相关服务的调用口与《SMC Calling Conventions》^[1]兼容。

飞腾系统服务调用接口（PSSI, Phytium System Service Interface）主要提供由于 PBF 的不同配置，可能动态变化的 SOC 信息，其余不随 PBF 配置而变化的信息，建议通过对应芯片的相关数据手册来获取。

4.1 飞腾系统服务调用约定

PBF 运行在 EL3，上层软件通过 AArch64 系统调用指令 SMC 调用本文档所描述的功能服务。

PBF 实现的自定义服务使用的 SMC 调用功能号与 SMC 调用规范兼容，即 SMC64: SiP Service Calls (0xC2000000-0xC200FFFF)。具体服务描述见表 4-1 所示。

表 4-1 飞腾系统服务总表

Name	Function ID	描述
PSSI_VERSION	0x8200FF03	PSSI 的接口规范版本
PBF_VERSION	0x82000001	PBF 固件版本号
CPU_VERSION	0xC2000002	处理器芯片版本
CPU_CORE_MAPS	0xC2000003	处理器核存在性位图
CPU_CORE_CONF	0xC2000004	处理器核配置信息

MEM_REGIONS	0xC2000005	内存地址空间
MCU_DIMMS	0xC2000006	内存条
PCI_CONTROLLER	0xC2000007	PCI 控制器
PCI_HOST_BRIDGE	0xC2000008	PCI 根桥
GET_SFW_BASE	0xC2000009	系统固件的基址
GET_CFGTBL_INFO	0xC200000A	系统配置表信息
GET_RESET_MODE	0xC200000B	系统复位原因
预留	0xC200000C	预留
预留	0xC200000D	预留
CPUFREQ_INFO	0xC200000E	查询处理器核频率信息
CPUFREQ_RW	0xC200000F	查询或设置处理器核的频率
SET_SUSPEND_TIMEOUT	0xC2000010	设置持续休眠的超时时间
SECURE_REG_RW	0xC2000011	安全寄存器读写
PFDI_REGISTER	0xC2000012	PFDI 服务函数注册
PFDI_DONE	0xC2000013	PFDI 服务完成
PBF_VERSION_DESCR	0xC2000014	PBF 固件版本描述
GET_PARAMETER_VERSION	0xC2000F00	获取支持参数表最大版本号
GET_RST_SOURCE	0xC2000F01	获取 CPU 复位原因
PLL_INIT*	0xC2000F02	PLL 初始化

PCIE_INIT	0xC2000F03	PCIE 初始化
DDR_INIT_SERVICES*	0xC2000F04	DDR 初始化
RELOCATE*	0xC2000F05	PBF 重定位
DEBUG_INIT	0xC2000F06	调试服务接口
SECURITY_CFG	0xC2000F07	CORE 安全等级配置

飞腾 PEI 服务使用注意事项：

- 1) *号表示的服务，只能在 RELOCATE 服务调用前使用。
- 2) RELOCATE 必须要调用，且只能在 DDR_INIT 服务成功后。否则 PBF 功能不完善。
- 3) 在 RELOCATE 服务成功前，PBF 仅能提供 PEI 服务，不支持其他服务。
- 4) 服务输入参数（x0-x3）未定义的寄存器，请保持值为 0。

返回的错误码为 64 位有符号整数，如表 4-2 所示。

表 4-2 PBF 服务调用错误码

返回值	错误含义	备注
0	成功调用	
-1	不支持该服务	
-2	无效参数	x1、x2、x3 分别返回对应调用输入参数的期望值
-3	DENIED	
-4	ALREADY_ON	
-5	ON_PENDING	

-6	INTERNAL_FAILURE	
-7	NOT_PRESENT	
-8	DISABLED	
-9	INVALID_ADDRESS	
-10	初始化失败	失败原因见 x1、x2、x3 返回值
-11	服务异常	

需要注意的是，PBF 阶段的所有数据访问都是 non-cacheable、物理地址访问模式，调用 PBF 服务时，如需要通过内存空间传递数据，须遵守该规则。

4.2 飞腾系统服务

4.2.1 PSSI_VERSION

该接口用于获取 PBF 提供的飞腾系统服务所遵循的接口规范的版本号。调用函数接口如下：

输入参数	x0: 功能号	PSSI_VERSION
返回值	x0	Bit[31:16]: 主版本号; Bit[15:0]: 次版本号

4.2.2 PBF_VERSION

该接口用于获取 PBF 固件的版本号。调用函数接口如下：

输入参数	x0: 功能号	PBF_VERSION
返回值	x0	Bit[31:16]: 主版本号;

		Bit[15:0]: 次版本号
--	--	-----------------

4.2.3 PBF_VERSION_DESCR

该接口用于获取 PBF 固件的版本的表述信息。函数调用接口如下：

输入参数	x0: 功能号	PBF_VERSION_DESCR
	x1: Buffer Address	返回 PBF 版本描述表的内存空间的物理地址
	x2: Buffer size	Buffer Address 指向的空间大小，以字节为单位
返回值	x0	0: 表示调用成功，返回内容格式参见表 4-3; -2: Buffer 大小不足，x2 返回期望的最小 buffer size
	x1	保留
	x2	调用参数 x2 给出 Buffer 大小不足时，返回期望的最小 buffer size

表 4-3 PBF 版本描述表

域	偏移	长度	说明
Build Date Offset	0	1	PBF Build Date 字符串在表中的偏移地址
Build Date String	Build Date Offset		PBF Build Date 描述字符串，以\0 结尾

4.2.4 CPU_VERSION

查询处理器 SOC 芯片的型号信息。调用函数接口如下：

输入参数	x0: 功能号	CPU_VERSION
	x1: CPU Socket ID	处理器的 Socket ID。单路系统中，该域为 0。
返回值	x0	0: 表示调用成功，处理器版本信息由 x1 返回； -1: 表示不支持该服务
	x1: cpu_type	处理器型号编码，编码含义见表 4-4。
	x3+x2: cpu_id	每颗处理器的唯一标识，共 128 位。x3 存放高 64 位，x2 存放低 64 位。

注：CPU_VERSION 的接口与《Phytium Base Firmware 接口规范》v2.4 及之前的版本不兼容。

每款飞腾处理器产品型号（cpu_type）都对应一个标准描述字符串，系统软件可以基于此字符串进行具体处理器型号相关的管理与控制。

表 4-4 飞腾处理器型号描述表

cpu_type	处理器型号描述字符串	描述
预留	预留	预留
0x000100	FT-2000A/2-X	FT-2000A/2
0x000101	FT-2000A/2	FT-2000A/2 标准版
预留	预留	预留
0x000200	FT-2000+/64-X	FT-2000+/64
0x000201	FT-2000+/64	FT-2000+/64 标准版
0x000202	FT-2000+/64-TY	FT-2000+/64 1.8GHz
0x000203	FT-2000+/64-TY32	FT-2000+/64 1.8GHz 32 核
0x000300	FT-2000/4-X	FT-2000/4
0x000301	FT-2000/4	FT-2000/4 标准版

0x000302	FT-2000/4-S	FT-2000/4 标准安全版
0x000303	FT-2000/4-I	FT-2000/4 工业级版
0x000304	FT-2000/4-SI	FT-2000/4 工业级安全版
0x000305	FT-2000/4-2	FT-2000/4 标准双核版
0x000306	FT-2000/4-I2	FT-2000/4 工业级双核版
0x000307	FT-2000/4-I1	FT-2000/4 工业级单核版
0x000308	FT-2000/4-M	FT-2000/4 移动版
0x000400	S2500/64	腾云 S2500/64
0x000401	S2500/64 C00	腾云 S2500/64 商业版
0x000402	S2500/64 I00	腾云 S2500/64 工业版
0x000403	S2500/56 C00	腾云 S2500/56 商业版
0x000404	S2500/56 I00	腾云 S2500/56 工业版
0x000405	S2500/48 C00	腾云 S2500/48 商业版
0x000406	S2500/48 I00	腾云 S2500/48 工业版
0x000500	D2000/8	腾锐 D2000/8
0x000501	D2000/8 B8C	腾锐 D2000/8 睿频版
0x000502	D2000/8 E8C	腾锐 D2000/8 标准版
0x000503	D2000/8 S8C	腾锐 D2000/8 标准网安版
0x000504	D2000/8 S8I	腾锐 D2000/8 工业级网安版
0x000505	D2000/4 S4C	腾锐 D2000/4 网安版
0x000506	D2000/4 S4I	腾锐 D2000/4 工业级网安版
0x000600	S5000	腾云 S5000
0x000700	E2000	腾珑 E2000

4.2.5 CPU_CORE_MAPS

获得处理器核的存在信息。调用函数接口如下：

输入参数	x0: 功能号	CPU_CORE_MAPS
	x1: Buffer address	处理器核存在信息存放地址
	x2: Buffer size	处理器核存在信息存放空间大小
返回值	x0	0: 表示调用成功, 处理器核存在信息描述见表 4-5; -2: Buffer 大小不足, x2 返回期望的最小 buffer size

表 4-5 处理器核存在描述表

名称	长度(byte)	偏移(byte)	说明
cpu_core_map_count	8	0x0	存在描述位图的个数
cpu_core_map_0	8	0x8	Core 63-0 的存在描述位图。 Bit 0 对应 core 0, bit 1 对应 core 1,, bit 63 对应 core 63。 每位: 1 表示存在, 0 表示不存在
cpu_core_map_1	8	0x10	Core 127-64 的存在描述位图。 格式和含义同上。
.....

4.2.6 CPU_CORE_CONF

查询处理器的配置信息。调用函数接口如下：

输入参数	x0: 功能号	CPU_CORE_CONF
------	---------	---------------

	x1: Buffer address	CPU 配置信息存放地址
	x2: Buffer size	CPU 配置信息存放空间大小
返回值	x0	0: 表示调用成功, 处理器数据描述见表 4-6; -2: Buffer 大小不足, x2 返回期望的最小 buffer size

表 4-6 处理器配置描述表

名称	长度(byte)	偏移(byte)	说明
cpu_freq	8	0x0	Core frequency (Hz)
cpu_l3c	8	0x8	L3 Cache size (Byte)
L3_cache_line	8	0x10	L3 Cache Line Size (Byte)

L1 和 L2 的相关信息, 可以通过体系结构系统寄存器来获得。

4.2.7 MEM_REGIONS

查询可用的内存地址空间信息。调用函数接口如下:

	x0: 功能号	MEM_REGIONS
输入参数	x1: Buffer address	内存地址空间信息存放地址
	x2: Buffer size	内存地址空间信息存放空间大小
返回值	x0	0: 表示调用成功, Memory 数据描述如表 4-7; -2: Buffer 大小不足, x2 返回期望的最小 buffer size

表 4-7 内存地址空间描述表

名称	长度(byte)	偏移(byte)	说明
mb_count	8	0x0	有效 memory block 数目
mem_blocks	-	0x8	Memory block 描述数组。每个数组项对应 1 个 memory block, 格式如表 4-8 所示

表 4-8 内存地址块

名称	长度(byte)	偏移(byte)	说明
mb_start	8	0x0	Memory block 起始地址
mb_size	8	0x8	Memory block 大小
mb_node_id	8	0x10	Memory block affinity 节点号

4.2.8 MCU_DIMMS

查询 MCU DIMM 相关信息。调用函数接口如下：

输入参数	x0: 功能号	MCU_DIMMS
	x1: Buffer address	MCU DIMM 信息存放地址
	x2: Buffer size	MCU DIMM 信息存放空间大小, 字节为单位
返回值		0: 表示调用成功, Memory 数据描述如表 4-9; -2: Buffer 大小不足, x2 返回期望的最小 buffer size

表 4-9 MCU DIMM 信息描述表

名称	长度(byte)	偏移(byte)	说明
mem_freq	8	0x0	频率 (MHz)
db_count	8	0x8	有效 mcu dimm block 数目
Dimm blocks	-	0x10	Mcu dimm block 数组。每个 MCU DIMM 对应一个如表 4-10 的描述块。

表 4-10 MCU DIMM 描述块

名称	长度(byte)	偏移(byte)	说明
db_capacity	8	0x0	MCU 内存总容量(MB)
db_dram_manufacturer_id	8	0x8	MCU 颗粒厂商 ID
db_module_manufacturer_id	8	0x10	MCU 模组厂商 ID
db_module_serial_number	8	0x18	MCU 模组序列号
db_slot_number	8	0x20	MCU 内存条个数。0, 表示该 MCU 没有插内存条, 本 MCU DIMM 描述块其它域都无效。
features	8	0x28	DIMM 的特征信息。 Bit 0: ECC Enable。 1: ECC 已使能; 0: ECC 未使能或不支持。 其它位保留

4.2.9 PCI_CONTROLLER

查询 PCI 控制器的相关信息。调用函数接口如下：

输入参数	x0: 功能号	PCI_CONTROLLER
	x1: Buffer base	信息存放地址
	x2: Buffer size	信息存放空间大小，字节为单位
返回值	x0	<p>0: 成功，buffer 中返回所有 PCI-E 控制器的描述，数据结构如表 4-11。</p> <p>-2: Buffer 大小不足，x2 返回期望的最小 buffer size</p>

表 4-11 PCI 控制器描述表

Field	Byte Length	Byte Offset	Description
count	8	0x0	控制器描述块个数
blocks	-	0x8	控制器描述块数组。每个控制器对应一个描述块。描述块的格式如表 4-12 所示。

表 4-12 控制器描述块

Field	Byte Length	Byte Offset	Description
lanes	1	0x0	控制器的 lane 数
speed	1	0x1	<p>1: Gen1;</p> <p>2: Gen2;</p> <p>3: Gen3;</p>

			4: Gen4; 5: Gen5; 其它值: 预留
Reserved	6	0x2	预留

4.2.10 PCI_HOST_BRIDGE

该接口用于查询 PCI Host Bridge 的相关信息。调用函数接口如下：

输入参数	x0	PCI_HOST_BRIDGE
	x1	Buffer base
	x2	Buffer size
返回值	x0	0: 成功, buffer 中返回所有 PCI-E Host Bridge 的描述, 数据结构如表 4-13 -2: Buffer 大小不足, x2 返回期望的最小 buffer size

表 4-13 PCI-E Host Bridge 描述结构

Field	Byte Length	Byte Offset	Description
hb_count	8	0x00	Host bridge count
hb_blocks	-	0x08	Host Bridge 描述块数组, 每个 Host Bridge 对应一个描述块, 描述块的格式如表 4-14。

每个 host bridge 对应一个 Host Bridge 描述块, 其格式如表 4-14 所示。其中, io_base、io_size、mem32_base、mem32_size、mem64_base、mem64_size 等值都

只是建议上层固件按该值划分对应的地址空间，上层可以结合芯片数据手册和实际平台需要，灵活分配对应地址空间的划分方式。

表 4-14 Host Bridge 描述块

Field	Byte Length	Byte Offset	Description
bus_start	1	0	起始总线号
bus_end	1	1	结束总线号
resv	6	2	reserved
cfg_base	8	8	PCI 配置空间基址
io_base	8	16	建议的 IO 空间基址
io_size	8	24	建议的 IO 空间大小
mem32_base	8	32	建议的 Mem32 空间基址
mem32_size	8	40	建议的 Mem32 空间大小
mem64_base	8	48	建议的 Mem64 空间基址
mem64_size	8	56	建议的 Mem64 空间大小
IntA	2	64	IntA 中断号
IntB	2	66	IntB 中断号
IntC	2	68	IntC 中断号
IntD	2	70	IntD 中断号

4.2.11 GET_SFW_BASE

该接口用于获取系统固件（System Firmware），比如，UEFI，在 Flash 中的

基址。调用函数接口如下：

输入参数	x0: 功能号	GET_SFW_BASE
返回值	x0	0: 表示调用成功; 其它值: 保留
	x1	Non-Secure Firmware 在 Flash 中的基址

4.2.12 GET_CFGTBL_INFO

该接口用于获取系统参数配置表(System Configuration Table)在 Flash 中的基址。调用函数接口如下：

输入参数	x0: 功能号	GET_CFGTBL_INFO
返回值	x0	0: 表示调用成功; -7: 表示不支持系统配置表; 其它值: 保留
	x1	系统配置表在 Flash 中的基址

4.2.13 GET_RESET_MODE

该接口用于查询系统复位的原因。其中，S3 是指 ACPI 定义的系统休眠到内存 (Suspend to RAM)。调用函数接口如下：

输入参数	x0: 功能号	GET_RESET_MODE
返回值	x0	0: 表示调用成功; -7: 表示不支持; 其它值: 保留

	x1	系统复位的原因： 0x55：从 S3 正常唤醒； 0xAA：从 S3 超时唤醒； 其它值：正常上电启动
--	----	--------------------------------------------------------------

4.2.14 SET_SUSPEND_TIMEOUT

系统 S3 之前，可以设置一个超时定时器，当该定时器超时，自动唤醒系统。该接口用于设置该定时器的超时值，也即系统在 S3 的持续时间，以分钟为单位。如果用户设置了这个值，当进入 S3 后，PBF 会启动超时定时器，定时器超时后，自动唤醒系统，并置系统复位模式值为 0xAA。如果要取消定时唤醒机制，需要将超时时间设置为 0。

调用函数接口如下：

输入参数	x0: 功能号	SET_SUSPEND_TIMEOUT
	x1: 超时值	无符号整数，持续处于 S3 状态的超时时间，以分钟为单位。 0：永不超时。
返回值	x0	0：表示调用成功； -7：表示不支持； 其它值：保留

4.2.15 CPUFREQ_INFO

查询处理器支持的频率信息，包括处理器核之间的动态调频关联关系和支持的频率集合。

与处理器实现相关，动态调节处理器核频率时，处理器核之间可能彼此相关（affected）。调频关联的处理器核构成关联组，调节组内某个核的频率，将对组内所有核产生相同的作用。关联组内的处理器核的逻辑号保持连续。

调用函数接口如下：

输入参数	x0: 功能号	CPUFREQ_INFO
	x1: Buffer address	处理器核频率信息存放地址
	x2: Buffer size	处理器核频率信息存放空间大小
返回值	x0	0: 表示调用成功, 处理器核存在信息描述见表 4-15; -2: Buffer 大小不足, x2 返回期望的最小 buffer size

表 4-15 处理器核频率信息表

名称	长度(byte)	偏移(byte)	说明
freq_count	2	0x0	频率点个数
aff_group_count	2	0x2	关联组个数
transition_latency	4	0x4	频率转换延迟, 以 ns (10^{-9} 秒) 为单位。
freqs[]	4* freq_count	0x8	支持的频率数组, 数组项个数由 freq_count 决定。每一项为对应的频率, 用 4 字节表示, 以 KHZ 为单位。
cores_in_group[]	2*aff_group_count	0x8 + 0x4* freq_count	每个调频关联组中处理器核的个数描述数组, 数组项个数由 aff_group_count 决定。每一项为组中的处理器核的个数, 用 2 字节表示。

4.2.16 CPUFREQ_RW

查询或设置指定处理器核的当前工作频率。调用函数接口如下：

输入参数	x0: 功能号	CPUFREQ_RW
	x1: operation	操作类型。 0: 查询; 1: 设置; 其它值: 保留
	x2: cpu_id	目标处理器核的标识, 即目标核的 MPIDR_EL1 值
	x3: target_freq_index	设置的目标频率在频率集合数组 freqs[]中的表项索引。该参数只有当 x1 表示设置操作时, 才有效。
返回值	x0	0: 表示调用成功, -1: 不支持该服务 -2: 非法调用参数
	x1	成功时 (x0 返回值为 0): <ul style="list-style-type: none"> ● 查询: 返回当前频率值, 以 KHZ 为单位。 ● 其它操作: 保留 调用参数非法 (x0 返回值为-2): 返回非法参数序号 <ul style="list-style-type: none"> ● 1: x1 非法 ● 2: x2 非法 ● 3: x3 非法 ● 其它值: 保留

4.2.17 SECURE_REG_RW

读写 Secure 属性的平台寄存器。

输入参数	x0: 功能号	SECURE_REG_RW
	x1: operation	操作类型。 0: 查询; 1: 设置; 其它值: 保留
	x2: address	目标平台寄存器的物理地址。
	x3: value	设置的值。该参数只有当 x1 表示设置操作时, 才有效。
返回值	x0	0: 表示调用成功, -1: 不支持该服务 -2: 非法调用参数
	x1	成功时 (x0 返回值为 0): <ul style="list-style-type: none"> ● 查询: 返回寄存器的当前值。 ● 其它操作: 保留 调用参数非法 (x0 返回值为-2): 返回非法参数序号 <ul style="list-style-type: none"> ● 1: x1 非法 ● 2: x2 非法 ● 3: x3 非法

- 其它值：保留

4.2.18 PFDI_REGISTER

当前 PBF 软件栈，PSCI 由 PBF 实现。操作系统通过 SMC 调用 PBF 提供的 PSCI 服务，实现系统的电源管理，比如：关机、重启、休眠（Suspend to RAM，S3）和休眠恢复（Resume from S3）。系统固件对上述操作透明，无法参与相关的管理。为了支持系统固件参与系统功耗管理过程，PBF 引入功耗管理控制派发机制 PFDI（Phytium Firmware Dispatch Interface），如图 4-1 所示。

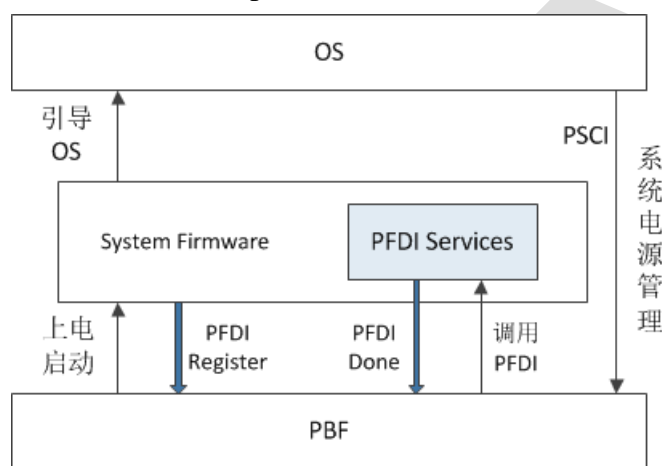


图 4-1 PFDI 机制

系统上电启动过程中，系统固件调用 PFDI_REGISTER 向 PBF 注册 PFDI 服务。当 OS 通过 PSCI 调用 PBF 系统电源管理服务时，PBF 调用系统固件注册的 PFDI 服务，PFDI 服务完成后，调用 PFDI_DONE 返回 PBF。另外，当系统从 S3 状态恢复进入 PBF 时，PBF 也会调用 PFDI 服务。如果系统固件未注册 PFDI 服务，则 PBF 在 PSCI 服务过程中并不会调用 PFDI 服务。

PBF 支持的 PFDI 服务如表 4-16 所示。其中，system_off_entry 和 system_reset_entry 分别在系统关机（SYSTEM_OFF）和系统重启（SYSTEM_RESET）。PBF 收到 SYSTEM_SUSPEND 服务请求后，首先调用 suspend_start_entry 通知系统固件要启动系统休眠，系统固件完成相关的服务操作后返回 PBF。PBF 完成休眠相关的操作后，再通过 suspend_end_entry 通知系统固件 PBF 完成了相应的休眠操作。系统从休眠状态恢复时，会调用 suspend_resume_entry 告知系统固件。

表 4-16 PFDI 服务表

PFDI 服务	说明	相关 PSCI 服务
system_off_entry	关机	SYSTEM_OFF
system_reset_entry	重启	SYSTEM_RESET
suspend_start_entry	启动休眠	SYSTEM_SUSPEND
suspend_end_entry	PBF 完成休眠相关操作	
suspend_resume_entry	休眠恢复	

为此，PBF 提供两个 PFDI 函数：PFDI_REGISTER、PFDI_DONE。

PFDI_REGISTER 的调用接口如下：

输入参数	x0: 功能号	PFDI_REGISTER
	x1: Table address	PFDI 服务函数表的物理地址。表格式见表 4-17。其中，域中填入的是对应服务函数的物理地址。如果不提供相应的服务，则对应域填 0。
返回值	x0	0: 表示调用成功， 其他值: 错误返回。

表 4-17 PFDI 服务函数表

域	偏移 (B)	长度 (B)
system_off_entry	0	8
system_reset_entry	8	8
suspend_start_entry	16	8
suspend_end_entry	24	8
suspend_resume_entry	32	8

需要注意的是，PBF 调用 PFDI 服务时，会将 PFDI 服务的执行环境配置为

系统固件调用 PFDI_REGISTER 时的环境。也就是说，PFDI 服务的执行环境（cache 是否打开、页表等）与系统固件调用 PFDI_REGISTER 时一致。另外，PFDI 服务的可用资源受限：

- 1、除 SP_EL2 外，所有系统寄存器都不可修改。
- 2、suspend_end_entry 服务不可访问内存。其它服务可用访问所有内存，但内存管理由系统固件自己负责，包括 PFDI 可能需要的预留内存。
- 3、可用访问所有的 IO 空间。

4.2.19 PFDI_DONE

系统固件在完成相应的 PFDI 服务后，调用 PFDI_DONE 返回 PBF。

输入参数	x0: 功能号	PFDI_DONE
返回值	x0: 调用错误状态	正常情况，PFDI_DONE 调用不会返回调用者。 如果返回，则 x0 返回错误原因。

4.2.20 GET_PARAMETER_VERSION

该接口用于获取当前支持的参数表最大版本号：

输入参数	x0: 功能号	GET_PARAMETER_VERSION
	x1	服务 ID: 0x00: PLL 参数表 0x01: PCIE 参数表 0x02: DDR 参数表 0x03: COMMON 参数表 其他值，未定义。
返回值	x0	Bit[31:16]: 主版本号; Bit[15:0]: 次版本号

4.2.21 GET_RST_SOURCE

该接口用于获取 CPU 复位原因。

输入参数	x0: 功能号	GET_RST_SOURCE
返回值	x0	Bit[31:0]: 复位原因 0x01: 上电复位 0x02: 软件热复位 (PLL 复位) 0x80: 看门狗复位 其他值: 保留

4.2.22 PLL_INIT

该接口用于初始化 SOC 频率。

输入参数	x0: 功能号	PLL_INIT
	x1	服务 ID: 0x00: 频率设置
	x2	PLL 服务参数表基地址 (格式见表 4-18)
返回值	x0	正确不返回。 错误见表 4-3。
	x1	错误参数编码 (仅仅错误返回为: -3 时有效): 0x1: CPU 频率错误。 0x2: DDR 频率错误。

使用此服务, 若设置正确会直接触发软件热复位 (PLL 复位) 去设置频率。

表 4-18 PLL 服务参数表

偏移地址	物理意义	宽度	说明
0x00_0000	PLL MAGIC	4	0x54460010
0x00_0004	PLL 参数表版本号	4	0x00000001
0x00_0008	PLL 参数表 SIZE	4	默认 0x100
0x00_000c	保留	4	
0x00_0010	CORE 频率	4	所有 core 都是此频率，单位为 MHz。比如，值为 2200，表示 2.2GHz。
0x00_0014	保留	4	
0x00_0018	LMU 频率	4	所有 DDR 都是此速率，单位为 4*Mbps。比如，值为 800，表示传输速率 3200Mbps。
0x00_001c	保留	4	
0x00_0020	保留	4	
0x00_0024	保留	4	
0x00_0028	保留	4	

4.2.23 PCIE_INIT

该接口用于初始化 PCIE 控制。

输入参数	x0: 功能号	PCIE_INIT
	x1	服务 ID: 0x00: PCIE 初始化服务
	x2	PCIE 服务参数表基地址（格式见表 4-19）
返回值	x0	正确返回 0，错误返回见表 4-3。
	x1	错误参数编码（仅仅错误返回为：-3 时有效）： 0x1: PCIE 0 初始化失败，丢失时钟。 0x2: PCIE 1 初始化失败，丢失时钟。

表 4-19 PCIE 服务参数表

偏移地址	物理意义	宽度	说明
------	------	----	----

0x00_0000	PCIE MAGIC	4	0x54460011
0x00_0004	PCIE 参数表版本号	4	0x00000002
0x00_0008	PCIE 参数表 SIZE	4	默认 0x100
0x00_000c	保留	4	
0x00_0014	PCIE 功能配置	4	bit[31:16] = 0, PCIE1 不初始化 1, CIE1 初始化为 x16 3, PEU1 初始化为 x8 其它值, 保留 bit[15:0] = 0, PCIE0 不初始化 1, PCIE0 初始化为 x16 3, PCIE0 初始化为 x8 其它值, 保留
0x00_0018	保留	16	
0x00_0028	PCIE0 控制器 0 基础配置	4	bit[23:16] = 0, EP 模式 1, RC 模式 其它值, 保留 bit[8:0] = 0, 自动协商, 不设置 1, 强制为 GEN1 2, 强制为 GEN2 3, 强制为 GEN3 其它值, 保留
0x00_002C	PCIE0 控制器 1 基础配置	4	同上
0x00_0030	PCIE0 控制器 2 基础配置	4	同上
0x00_0034	PCIE0 控制器 0 均衡值	4	bit[8:0] = N, PRESET N (值范围 0-10)
0x00_0038	PCIE0 控制器 1 均衡值	4	同上
0x00_003C	PCIE0 控制器 2 均衡值	4	同上
0x00_0040	保留	80	
0x00_0090	PCIE1 控制器 0 基础配置	4	bit[23:16] = 0, EP 模式 1, RC 模式 其它值, 保留 bit[8:0] = 0, 自动协商, 不设置 1, 强制为 GEN1 2, 强制为 GEN2 3, 强制为 GEN3 其它值, 保留
0x00_0094	PCIE1 控制器 1 基础配置	4	同上

0x00_0098	PCIE1 控制器 2 基础配置	4	同上
0x00_009C	PCIE1 控制器 0 均衡值	4	bit[8:0] = N, PRESET N (值范围 0-10)
0x00_00A0	PCIE1 控制器 1 均衡值	4	同上
0x00_00A4	PCIE1 控制器 2 均衡值	4	同上

4.2.24 DDR_INIT_SERVICES

该服务为功能服务组，使用相同的功能号，用各服务 ID 进行功能区分，各服务 ID 见表 4-20。

表 4-20 DDR 初始化服务表

服务 ID	说明
0x00	DDR 初始化服务
0x01	DDR 进入自刷新
0x02	DDR 自刷新锁定/解锁

4.2.24.1 DDR_INIT

该接口用于初始化 DDR 控制器。

输入参数	x0: 功能号	DDR_INIT
	x1	服务 ID: 0x00: DDR 初始化服务
	x2	DDR 服务参数表基地址 (格式见表 4-3)
返回值	x0	正确返回 0, 错误返回见表 4-21。
	x1	错误参数编码 (仅仅错误返回为: -10 时有效): Bit[31:24]: 出错的通道 每一个 bit 代表一个通道, bit24 对应通道

		0, bit25 对应通道 1, 依此类推; Bit[23:0]: 错误编码 0x1: training fail 其他值: 保留
--	--	----------------------------------------------------------------------------------------

表 4-21 DDR 服务参数表

偏移地址	物理意义	宽度	说明
0x00_0000	DDR MAGIC	4	0x54460014
0x00_0004	DDR 参数表版本号	4	0x00000001
0x00_0008	DDR 参数表 SIZE	4	默认 0x100
0x00_000c	保留	4	
0x00_0010	通道控制	1	BIT 0: CH0 使能位 1: 打开通道 0 0: 关闭通道 0 BIT 1: CH1 使能位 1: 打开通道 1 0: 关闭通道 1 BIT [7:2]: 保留
0x00_0011	ECC 控制	1	BIT0: CH0 DDR ECC 使能位 1: DDR 内存模块支持 ECC, 则使能 ECC 0: 不使能 ECC BIT1: CH1 DDR ECC 使能位 1: DDR 内存模块支持 ECC, 则使能 ECC 0: 不使能 ECC BIT [7:2]: 保留
0x00_0012	DM 控制	1	BIT0: CH0 DM 使能位 1: 使能 DM 0: 不使能 DM BIT1: CH1 DM 使能位 1: 使能 DM 0: 不使能 DM BIT [7:2]: 保留
0x00_0013	保留	1	
0x00_0014	MCU 基础配置	1	BIT0: 读取 SPD 频率使能 1: 读取 SPD 频率 从 SPD 中读取频率, 对比 PLL 参数表的 DDR 传输速率, 取他们的最小值作为最终的 DDR 传输速率,

			<p>单位为 4*Mbps。</p> <p>0: 不读 SPD 频率</p> <p>BIT1: 2D training 时, 是否采用 margin 值判断 training 是否通过</p> <p>1: 采用 margin 判断</p> <p>0: 不采用 margin 判断</p> <p>BIT2: S3 回来时 mcu 是否采用 devinit 模式</p> <p>1: 采用 devinit 模式</p> <p>0: 采用正常模式</p> <p>BIT3: 2T 模式控制</p> <p>1: 使能 2T 模式</p> <p>0: 默认 1T 模式</p> <p>BIT4: 一拖二使能</p> <p>1: 使能一拖二, 一个通道插两根内存条</p> <p>0: 禁用一拖二, 一个通道插一根内存条</p> <p>其他位, 保留</p>
0x00_0015	Training debug 控制	1	<p>Training debug 控制:</p> <p>0x04: full debug</p> <p>0x0a: coarse debug</p> <p>0xc8: stage completion</p> <p>0xc9: assertion messages</p> <p>0xff: only completion</p> <p>其他值: 保留</p>
0x00_0016	Training recover	1	<p>Training recover 控制:</p> <p>0: phyinit 时正常 training</p> <p>1: phyinit 时不做 training, 而是从 flash 中读取之前保存的配置数据</p> <p>其他值: 保留</p>
0x00_0017	保留	9	
0x00_0020	通道 0 DDR 信息	64	<p>BYTE 0x0: dimm type</p> <p>1: RDIMM 2: UDIMM</p> <p>3: SODIMM 4: LRDIMM</p> <p>BYTE 0x1: data width</p> <p>0: X4 1: X8</p> <p>2: X16 3: X32</p> <p>BYTE 0x2: mirror type</p> <p>0: standard 1: mirror</p> <p>BYTE 0x3: ecc type</p> <p>0: no ecc 1: ecc</p> <p>BYTE 0x4: 保留</p> <p>BYTE 0x5: rank num</p>

			0x1: 1 rank 0x2: 2 rank 0x4: 4 rank BYTE 0x6: row num BYTE 0x7: column num BYTE 0x8: bank group num BYTE 0x9: bank num BYTE 0xa-0xb: module manufacturer id BYTE 0xc-0xd: tAAmin BYTE 0xe-0xf: tRCDmin BYTE 0x10-0x11: tRPmin BYTE 0x12-0x13: tRASmin BYTE 0x14-0x15: tRCmin BYTE 0x16-0x17: tFAWmin BYTE 0x18-0x19: tRRD_Smin BYTE 0x1a-0x1b: tRRD_Lmin BYTE 0x1c-0x1d: tCCD_Lmin BYTE 0x1e-0x1f: tWR_min BYTE 0x20-0x21: tWTR_Smin BYTE 0x22-0x23: tWTR_Lmin BYTE 0x24-0x3f: 保留
0x00_0060	通道 1 DDR 信息	64	同上

4.2.24.2 DDR_SUSPEND_ENTRY

该接口用让内存进入自刷新。

输入参数	x0: 功能号	DDR_INIT
	x1	服务 ID: 0x01: DDR 进入自刷新
返回值	x0	正确返回 0, 错误返回见表 4-3。

4.2.24.3 DDR_SUSPEND_LOCK/UNLOCK

该接口用让内存进入自刷新后且 VTT 电源稳定后, 锁定状态。

输入参数	x0: 功能号	DDR_INIT
	x1	服务 ID:

		0x02: DDR 自刷新锁定 0x03: DDR 自刷新解锁
返回值	x0	正确返回 0, 错误返回见表 4-3。

4.2.25 RELOCATE

该服务用于重定位 PBF, 开启所有 PBF 服务。该服务很重要, 一定要被调用, 且必须在内存初始化成功后才能调用。

输入参数	x0: 功能号	RELOCATE
	x1	服务 ID: 0x00: 上电服务 0x01: 休眠恢复服务
	x2	PBF 基础服务参数表基地址 (格式见表 4-22)
返回值	x0	正确返回 0, 错误返回见表 4-3。
	x1	错误编码未定

表 4-22 PBF 基础服务参数表

偏移地址	物理意义	宽度	说明
0x00_0000	COMMON MAGIC	4	0x54460013
0x00_0004	COMMON 参数表版本号	4	0x00000001
0x00_0008	COMMON 参数表 SIZE	4	默认 0x100
0x00_000c	保留	4	
0x00_0010	CORE 的使能 BITMAPs	8	每个 bit 代表一个 core。 Bit 0 对应 core 0: 1: 表示使能 0: 表示关闭 Bit 1 表示 core1, 以此类推

4.2.26 DEBUG_INIT

该服务控制 PBF 调试打印等级。

输入参数	x0: 功能号	DEBUG_INIT
	x1	服务 ID: 0x00: 调试打印等级设置
	x2	服务等级 (默认 0x0): 0x0: 关闭调试打印 0x1: 一级 0x2: 二级 其他值保留。
返回值	x0	正确返回 0, 错误返回见表 4-3。

4.2.27 SECURITY_CFG

该服务控制 CORE 的安全等级。

输入参数	x0: 功能号	SECURITY_CFG
	x1	服务 ID: 0x00: 配置安全等级
	x2	服务等级 (默认 0x0): 0x0: level 0, 默认级别 0x1: level 1, 中安全级别, 防 2/3/4, 不防 1 0x2: level 2, 高安全级别, 防 1/2/3/4 其他值保留。
返回值	x0	正确返回 0, 错误返回见表 4-3。

5 飞腾 LOGO 的固件使用规则

飞腾官方 LOGO 如图 5-1 所示。



图 5-1 飞腾 LOGO

飞腾系统上，固件在显示开机画面时，须在显著位置显示飞腾 LOGO。图 5-2 是飞腾系统的开机画面举例示意图，在整机开机画面的基础上，右上角增加飞腾 LOGO。



图 5-2 固件使用飞腾 LOGO 示意图

参考文献

[1] SMC Calling Conventions (ARM DEN 0028).

[2] ARM Power State Coordination Interface (ARM DEN 0022).

[3] ARM System Control and Management Interface Platform Design Document (ARM DEN0056A)

PHYTIUM