

Document Title: PM_FW device driver module design
description of Safety Control System

Document Number: 17-Q03-000512

Project Number: CT-RD-1601

Project Name: First phase of Safety Control System
Development Project

Material Number: N/A

Document Version: A

Classification Level: Highly secret

Document Status: CFC

Controlled Status: Under control

Prepared by: Zhang Lei 2016-12-27

Checked by: Zhu Genghua 2016-12-30

Countersigned by: Liu Yue

Approved by: Wen Yiming 2016-12-30

Revision History

No.	Relevant Chapter	Change Description	Date	Version Before Change	Version After Change	Prepared by	Checked by	Approved by
1		Document created	2016-12-27	None	A	Zhang Lei	Zhu Genghua	Wen Yiming
2								
3								
4								
5								

Relationship between this version and old versions: None.

文件名称：安全控制系统 PM_FW 设备驱动模块设计说明书

文件编号：17-Q03-000512

项目编号：CT-RD-1601

项目名称：安全控制系统开发项目一期

物料编号：

版本号/修改码：A

文件密级：机密

文件状态：CFC

受控标识：受控

拟制：张磊

2016 年 12 月 27 日

审核：朱耿华

2016 年 12 月 30 日

会签：刘跃

批准：温宜明

2016 年 12 月 30 日

修订页

编号	章节名称	修订内容简述	修订日期	订前版本	订后版本	拟制	审核	批准
1		创建	2016-12-27		A	张磊	朱耿华	温宜明
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

本版本与旧文件（版本）的关系：

Content 目录

1	Document overview 文档概述.....	1
1.1	Introduction 综述	1
1.2	Reference 参考文档.....	1
1.2.1	Project documents 内部参考文档	1
1.3	Terms and abbreviations 术语和缩略语	1
1.3.1	Terms 术语	1
1.3.2	Abbreviations 缩略语	2
2	Module overview 模块概述.....	3
3	Module design 模块设计	4
3.1	Function description 功能描述	4
3.2	Design concept 设计思路	4
3.3	Interface function 接口函数.....	4
3.4	Global variable 全局变量	9
3.5	Data structure 数据结构.....	10
3.6	List of sub-function 子功能列表	10
4	Design of sub-function 子功能设计	11
4.1	CM_BUS driver CM_BUS 驱动.....	11
4.1.1	Copy data to CM_BUS buffer	11
4.1.2	Start CM_BUS transmit	12
4.1.3	Check CM_BUS send status	13
4.1.4	Check CM_BUS receive status	13
4.1.5	Copy data from CM_BUS buffer	14
4.2	PM_BUS driver PM_BUS 驱动	15
4.2.1	Set PM_BUS slot number	15
4.2.2	Copy data to PM_BUS buffer	15
4.2.3	Copy data from PM_BUS buffer.....	16
4.2.4	Start PM_BUS transmit.....	17
4.2.5	Check PM_BUS receive status.....	18
4.2.6	Check PM_BUS send status.....	18
4.3	IP_BUS driver IP_BUS 驱动	19
4.3.1	IP_BUS send data.....	19
4.3.2	IP_BUS receive data	20
4.4	PCIe driver PCIe 驱动.....	21
4.4.1	PCIe send data.....	21

4.4.2	PCIe receive data.....	22
4.5	SPI driver SPI 驱动	22
4.5.1	Read SPI.....	22
4.5.2	Write SPI	23
4.6	Nor Flash driver Nor Flash 驱动.....	24
4.6.1	Erase NOR Flash	24
4.6.2	Write NOR Flash.....	25
4.6.3	Read NOR Flash.....	26
4.7	SPI Flash driver SPI Flash 驱动.....	27
4.7.1	Read data from SPI Flash.....	27
4.7.2	Write data to SPI Flash.....	28
4.7.3	Erase SPI Flash.....	29
4.8	FPGA driver FPGA 驱动	30
4.8.1	Read FPGA	30
4.8.2	Write FPGA.....	31
4.9	DFS driver DFS 驱动	32
4.9.1	DFS Read	32
4.9.2	DFS Write.....	33
4.9.3	DFS Erase.....	34
4.9.4	DFS Initialize	34
4.9.5	Register device to DFS.....	35
4.9.6	Deregister device from DFS	36

1 Document overview 文档概述

1.1 Introduction 综述

This document describes the design description of device driver of PM_FW of Safety Control System. The document describes the overall concept of the function of the module, and then the sub-function of the modules are described in detail.

This document is the output of module design phase of PM device driver, and is the input for the follow-up coding phase.

本文档描述了安全控制系统中 PM 设备驱动的设计方案。文档首先描述了模块功能的总体设计思路，然后将模块功能划分为若干子功能并进行详细说明。

本文档是 PM 设备驱动模块设计的输出，也是后续编码的输入。

1.2 Reference 参考文档

1.2.1 Project documents 内部参考文档

[1] Embedded software safety concept of Safety Control System [505], 15-Q02-000059

[1] 安全控制系统嵌入式软件安全概念说明书 [505], 15-Q02-000059

[2] PM_FW software overall design description of safety control system [506], 15-Q02-000074

[2] 安全控制系统 PM_FW 总体设计说明书 [506], 15-Q02-000074

1.3 Terms and abbreviations 术语和缩略语

1.3.1 Terms 术语

Table 1-1 Terms

表 1-1 术语

No. 序号	Term 术语	Description 解释
1.	IP_BUS	Communication between PM and IO modules. PM 与 IO 模块之间的通讯总线。
2.	CM_BUS	Communication between PM and CM. PM 与 CM 之间的通讯总线。
3.	PM_BUS	Communication between PMs. PM 之间的通讯总线。
4.	System Net	Communication between control station and PC. 控制站与上位机之间的通讯网络。
5.	Safety Net	Safe communication between control stations.

		控制站之间的安全通讯。
6.	Control station 控制站	A set of triple redundant control system, which includes triple redundant PMs and IO modules under control. 一套三冗余的控制系统，包含三冗余 PM 和 PM 控制的各种 IO 模块。
7.	System response time 系统响应时间	Time interval from the moment that transition of demand signal generated at input ETP to the moment that transition of response signal generated at output ETP. 从系统输入端子板上产生需求信号跳变的时刻到输出端子板上产生相应的响应信号跳变之间的时间。
8.	Control cycle 控制周期	Time interval between adjacent two runs of user program execution. PM 两次执行用户程序间隔时间。
9.	Project 工程	Files which contain configuration information for control station and generated by IEC 61131 configuration software. These files contain all the information required by control station to implement control, including user control program (binaries) to be loaded and executed as well as configuration information of task, CM, PM and IO modules. IEC 61131 组态软件在完成编译后，为控制站生成的组态信息文件，该文件包含可加载执行的用户控制程序（二进制程序）、任务配置信息、CM 配置信息、PM 配置信息和 IO 模块配置信息等各种控制站完成控制所需的信息。
10.	Source project 源工程文件	Source file of the project before compiling. 工程在编译前的源文件。
11.	User program 用户程序	Part of project which contain user control program (binaries) to be loaded and executed and configuration information of task. 工程中的一部分：可加载执行的用户控制程序（二进制程序）和任务配置信息。

1.3.2 Abbreviations 缩略语

Table 1-2 Abbreviations

表 1-2 缩略语

No. 序号	Abbreviation 缩略语	English description 英文	Chinese description 中文
1.	PM	Processor Module	主处理器模块
2.	CM	Communication Module	通讯模块
3.	BI	Bus Interface Module	总线接口模块
4.	AI	Analog Input Module	模拟量输入模块
5.	AO	Analog Output Module	模拟量输出模块

6.	DI	Digital Input Module	数字量输入模块
7.	DO	Digital Output Module	数字量输出模块
8.	OSP	Over Speed Protect Module	超速保护模块
9.	SOE	Sequence Of Events	SOE 事件
10.	SIL	Safety Integrity Level	安全完整等级
11.	PW	Power Module	电源模块
12.	OPC	OLE for Process Control	用于过程控制的对象链接与嵌入式技术
13.	UP	User Program	用户程序
14.	DFS	Device file system	设备文件系统

2 Module overview 模块概述

The location of the device driver module (marked red) in the software hierarchy is shown below.

设备驱动模块（标红）在软件层次中的位置如下图所示。

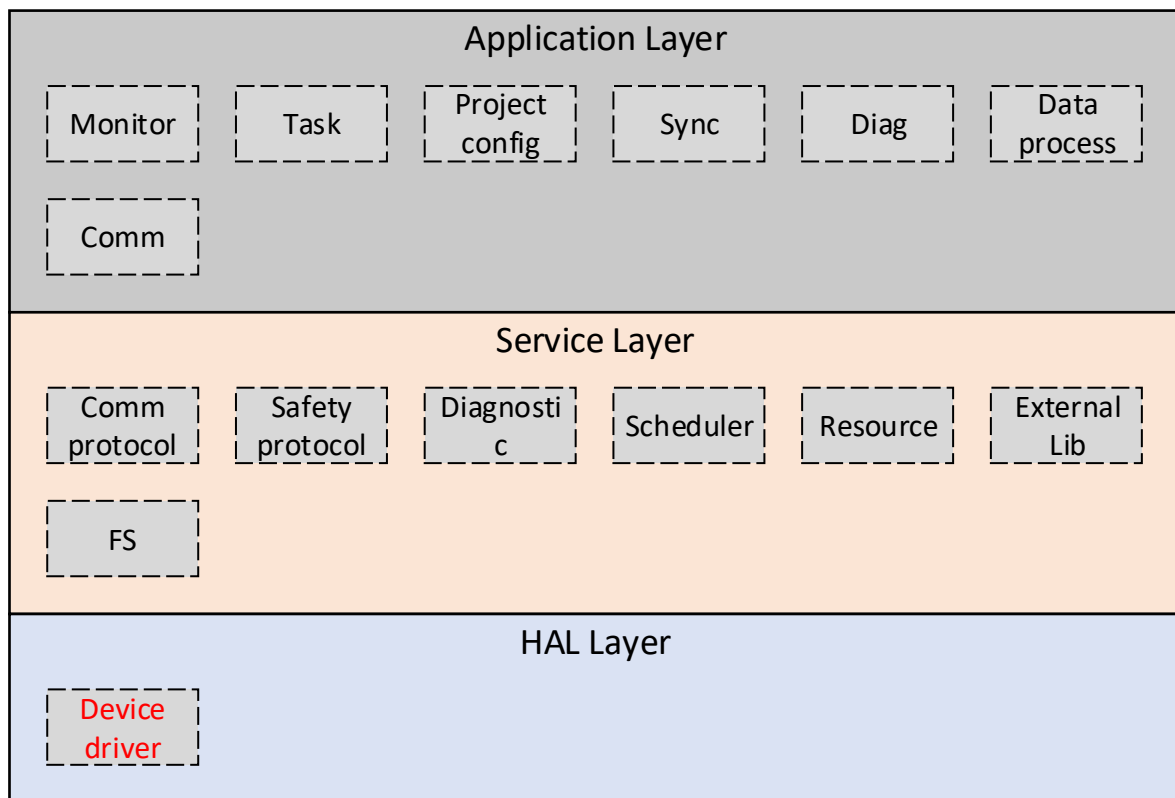


Figure 2-1 the location of the data processing module

图 2-1 模块位置

Device driver module is used to operate various kinds of peripherals, include various buses, Flash, FPGA, etc.

设备驱动模块主要用于操作各种外设，包括各种总线、Flash、FPGA 等。

3 Module design 模块设计

3.1 Function description 功能描述

This module realizes the drivers of various peripherals on PM: CM/PM/IP_BUS, SPI, Flash, PCIe, FPGA and so on. Various peripherals can be operated by the module.

本模块实现 PM 上各种外设的驱动程序：CM/PM/IP 总线、SPI、Flash、PCIe、FPGA 等。通过该模块可以操作各个外设。

3.2 Design concept 设计思路

Write the code according to the operation flow in the device's data sheet.

根据设备的数据手册中的操作流程编写代码。

3.3 Interface function 接口函数

The interface functions which is provided by this module is shown as follows:

模块提供的接口函数如下：

1. INT32 dfs_write(INT32 dfd, UINT32 dst,UINT8 *src,UINT32 cnt)

Input argument 输入参数	Output argument 输出参数	Description 描述
dfd - device's DFS ID. dst - destination data address. src - write data buffer. cnt - data length. dfd - 设备的 DFS ID。 src - 数据目标地址。 buf - 写数据缓存。 cnt - 数据长度	negative - operation failure. Other value is the length of the data actually written. 负数 - 操作失败。 其它值为实际写入的数据的长度。	This function is an abstract interface of write device. 本函数是写外设的抽象接口。

2. INT32 dfs_read(INT32 dfd, UINT32 src, UINT8 *buf,UINT32 cnt)

Input argument 输入参数	Output argument 输出参数	Description 描述
dfd - device's DFS	negative - operation	This function is an abstract

ID. src - source data address. buf - read data buffer. cnt - data length. dfd - 设备的 DFS ID. src - 数据源地址。 buf - 读数据缓存。 cnt - 数据长度	failure. Other value is the length of the data actually read. 负数 - 操作失败。 其它值为实际读出的数据的长度。	interface of read device. 本函数是读外设的抽象接口。
---	---	--

3. INT32 dfs_erase(INT32 dfd,UINT32 no,UINT32 num)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
dfd - device's DFS ID. no - erase from this block no. num - block number to be erase. dfd - 设备的 DFS ID。 no - 从该块号开始擦除。 num - 要擦除的块数。	0 - operation succeeds. negative - operation failure 0 - 操作成功。 负数 - 操作失败。	This function is an abstract interface of erase device. 本函数是擦除外设的抽象接口。

4. uint16_t fpga_read(uint32_t reg_num)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
reg_num - register number. reg_num - 寄存器号。	FPGA register's value. FPGA 寄存器的值。	This function is used to read FPGA register. 本函数用于读取 FPGA 寄存器。

5. void fpga_write(uint32_t reg_num,uint16_t value)

Input argument	Output argument	Description
----------------	-----------------	-------------

接口输入参数	接口输出参数	描述
reg_num - register number. value - new value of register. reg_num - 寄存器号。 value - 寄存器的新值。	No. 无。	This function is used to modify FPGA register. 本函数用于修改 FPGA 寄存器。

6. INT32 lld_write (ADDRESS addr, const UINT8 *src, UINT32 cnt)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
addr - object address. src - data address. cnt - data length. addr - 目标地址。 src - 数据地址。 cnt - 数据长度。	The length of the data is actually written. 实际写入的数据的长度。	This function is used write data to NOR Flash. 本函数用于将数据写入 NOR Flash。

7. INT32 lld_read (ADDRESS addr , UINT8* buf, UINT32 cnt)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
addr - source address. src - data address. cnt - data length. addr - 源地址。 src - 数据地址。 cnt - 数据长度。	The length of the data actually read. 实际读出的数据的长度。	This function is used to read data from NOR Flash. 本函数用于从 Flash 读出数据。

8. INT32 lld_erase(INT32 no, INT32 num)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
no - erase from this block no. num - block number	0 - operation succeeds. negative - operation	This function is used to erase NOR Flash. 本函数用于擦除 NOR Flash。

to be erase. no - 从该块号开始擦除。 num - 要擦除的块数。	failure 0 - 操作成功。 负数 - 操作失败。	
---	------------------------------------	--

9. void pcie_write(uint8_t *dst, uint8_t *src, uint32_t len)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
dst - destination offset address in PCIe region. src - source address in memory. len - data length. dst - PCIe 域中的目标偏移地址。 src - 内存上的源地址。 len - 数据长度。	Length of the send data. 发出数据的长度。	This function is used to send data over PCIe. 本函数用于通过 PCIe 发送数据。

10. void pcie_read(uint8_t *dst, uint8_t *src, uint32_t len)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
dst - destination address in memory. src - source offset address in PCIe region. len - data length. dst - 内存上的目标地址。 src - PCIe 域中的源偏移地址。 len - 数据长度。	No. 无。	This function is used to process the IO output data cyclically. 本函数用于周期进行 IO 数据输出处理。

11. INT32 spiflash_read(UINT32 addr, UINT8 *buf, UINT32 cnt)

Input argument	Output argument	Description
----------------	-----------------	-------------

接口输入参数	接口输出参数	描述
addr - Data address in flash. buf - Data buffer. cnt - Data length. addr - 数据在 Flash 上的地址。 buf - 数据缓存。 cnt - 数据长度。	negative - operation failure. The length of the data actually read. 负数 - 操作失败。 实际读出的数据的长度。	This function is used to read data from SPI Flash. 本函数用于从 SPI Flash 读取数据。

12. INT32 spiflash_write(UINT32 addr , UINT8 *buf , UINT32 cnt)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
addr - data address in flash. buf - source data buffer. cnt - data length. addr - 数据在 Flash 上的地址。 buf - 源数据缓存。 cnt - 数据长度。	negative - operation failure. The length of the data is actually written. 负数 - 操作失败。 实际写入的数据的长度。	This function is used write data to SPI Flash. 本函数用于把数据写到 SPI Flash。

13. INT32 spiflash_erase(INT32 no , INT32 num)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
no - erase from this block no. num - block number to be erase. no - 从该块号开始擦除。 num - 要擦除的块数。	0 - operation succeeds. negative - operation failure 0 - 操作成功。 负数 - 操作失败。	This function is used to erase SPI Flash. 本函数用于擦除 SPI Flash。

14. INT16 ipbus_write(UINT8 opt,UINT8 *data,UINT32 offset,UINT32 len)

Input argument	Output argument	Description
----------------	-----------------	-------------

接口输入参数	接口输出参数	描述
opt - send option. data - data to send offset -address of the data send to. len - data length. opt - 发送选项。 data - 要发送的数据。 offset - 数据发送到的地址。 len - 数据长度。	positive - send data's length. negative - operation failure 正数 - 发送的数据的长度。 负数 - 操作失败。	This function is used to send data over IP_BUS. 本函数用于在 IP_BUS 上发送数据。

15. INT16 ipbus_read(UINT8 opt, UINT8 *data, UINT32 offset, UINT32 len)

Input argument 接口输入参数	Output argument 接口输出参数	Description 描述
opt - receive option. data - store received data offset - address of the data received from. len - data length. opt - 接收选项。 data - 保存接收到的数据。 offset -接收到的数据的地址。 len - 数据长度。	positive - send data's length. negative - operation failure 正数 - 发送的数据的长度。 负数 - 操作失败。	This function is used to receive data from IP_BUS. 本函数用于从 IP_BUS 接收数据。

3.4 Global variable 全局变量

Table 3-1 Global variable list

表 3-1 全局变量列表

No. 序号	Type 变量类型	Name 名称	Description 描述
1.	struct dfs_ops	*dfs_table[DFS_NUM]	Record all device that register to DFS. 记录注册到 DFS 的所有设备。

2.	struct flash_struct	fpgaflashSt	Describe the construction of SPI Flash. 描述 SPI Flash 的结构。
3.	struct dfs_ops	fpgaflashOps	Describe SPI Flash's operation function. 描述 SPI Flash 的操作函数。
4.	INT32	dfd3	Record the DFS ID of SPI Flash. 记录 SPI Flash 的 DFS ID。
5.	struct flash_struct	flash_struct [3]	Record all Flash's construction. 记录所有 Flash 的结构。
6.	struct dfs_ops	flash_ops[3]	Record all Flash's operation function. 记录全部 Flash 的操作函数。

3.5 Data structure 数据结构

1. DFS descriptor

```
struct dfs_ops {
    INT32 (*write)(UINT32 dst,UINT8 *src,UINT32 cnt );
    INT32 (*read)(UINT32 src, UINT8 *buf, UINT32 cnt);
    INT32 (*erase)(INT32 no,INT32 num);
    INT32 (*lseek)(INT32 offset, INT32 whence);
    INT32 (*ioctl)(INT32 cmd, void *args);
    INT32 base_addr; //record flash's base address ,but may not use from this offset
    INT32 start;
    INT32 end;
    INT32 size;
    INT32 eraseUnit;
    INT32 *errno;
    UINT8 *dname;
    void *data;
};
```

2. Flash descriptor

```
struct flash_struct {
    INT32 seSize;
    INT32 seNum;
    INT32 ssSize;
    INT32 ssNum;
    INT32 totalSize;
};
```

3.6 List of sub-function 子功能列表

The sub-functions list is shown as follows:

子功能列表如下。

Table 3-2 Sub function list

表 3-2 子功能列表

Sub function No. 子功能编号	Function description 功能描述
SWDD-PM-DD_NSafR_NSecR_A_001	CM_BUS driver. CM_BUS 驱动。
SWDD-PM-DD_NSafR_NSecR_A_002	PM_BUS driver. PM_BUS 驱动。
SWDD-PM-DD_NSafR_NSecR_A_003	IP_BUS driver. IP_BUS 驱动。
SWDD-PM-DD_NSafR_NSecR_A_004	PCIe driver. PCIe 驱动。
SWDD-PM-DD_NSafR_NSecR_A_005	SPI driver. SPI 驱动。
SWDD-PM-DD_NSafR_NSecR_A_006	Nor Flash driver. Nor Flash 驱动。
SWDD-PM-DD_NSafR_NSecR_A_007	SPI Flash driver SPI Flash 驱动。
SWDD-PM-DD_NSafR_NSecR_A_008	FPGA driver. FPGA 驱动。
SWDD-PM-DD_NSafR_NSecR_A_009	Device file system driver 设备文件系统驱动。

4 Design of sub-function 子功能设计

4.1 CM_BUS driver CM_BUS 驱动

SWDD-PM-DD_NSafR_NSecR_A_001

This sub-function is used to transmit data over the CM_BUS.

本子功能用于在 CM_BUS 上传输数据。

4.1.1 Copy data to CM_BUS buffer

4.1.1.1 Function Description 功能描述

This function is used to copy user data to CM_BUS buffer.

该函数用于将用户数据拷贝到 CM_BUS 缓存。

4.1.1.2 Argument Description 参数说明

➤ Definition 函数定义

int16_t copy_to_cmbuf(uint8_t *data, int32_t len)

➤ Input argument 输入参数

data - data to copy.

len - data length.

data - 要拷贝的数据。

len - 数据长度

➤ Output argument 输出函数

positive - copied data's length.

negative - operation failure

正数 - 拷贝的数据的长度。

负数 - 操作失败。

4.1.1.3 Processing flow 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.1.2 Start CM_BUS transmit

4.1.2.1 Function Description 功能描述

This function starts CM_BUS transmit.

该函数发起 CM_BUS 传输。

4.1.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t enable_cm_send(INT32 station)

➤ Input argument 输入参数

station - target CM station number.

station - 目标 CM 站号。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.1.2.3 Processing flow 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.1.3 Check CM_BUS send status

4.1.3.1 Function Description 功能描述

This function is used to check the status of CM_BUS send.

该函数用来检查 CM_BUS 发送的状态。

4.1.3.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t check_cm_send(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure.

0 - 操作成功。

负数 - 操作失败。

4.1.3.3 Processing flow 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.1.4 Check CM_BUS receive status

4.1.4.1 Function Description 功能描述

This function is used to check the status of CM_BUS receive.

该函数用来检查 CM_BUS 接收的状态。

4.1.4.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t check_cm_recv(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.1.4.3 Processing flow 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.1.5 Copy data from CM_BUS buffer

4.1.5.1 Function Description 功能描述

This function is used to get data from CM_BUS buffer.

该函数用来从 CM_BUS 缓存获取数据。

4.1.5.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t copy_from_cmbuf(uint8_t *buf)

➤ Input argument 输入参数

buf - receive data buffer.

buf - 接收数据缓存。

➤ Output argument 输出函数

positive - received data's length.

negative - operation failure

正数 - 接收到的数据长度。

负数 - 操作失败。

4.1.5.3 Processing flow 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2 PM_BUS driver PM_BUS 驱动

SWDD-PM-DD_NSafR_NSecR_A_002

This sub-function is used to transmit data over the PM_BUS.

本子功能用于在 PM_BUS 上传输数据。

4.2.1 Set PM_BUS slot number

4.2.1.1 Function Description 功能描述

This function is used to set PM_BUS' station number.

本函数用于设置 PM_BUS 的站号。

4.2.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

int8_t set_pm_slot(void)

➤ Input argument 输入参数

No.

无。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.2.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2.2 Copy data to PM_BUS buffer

4.2.2.1 Function Description 功能描述

This function is used to copy data to PM_BUS buffer.

本函数用于将数据拷贝到 PM_BUS 缓存。

4.2.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t copy_to_pmbuf(int8_t bus_no, uint8_t *data, int32_t len)

➤ Input argument 输入参数

bus_no - PM_BUS' bus no.

data - data to copy.

len - data length.

bus_no - PM_BUS 的总线号。

data - 要拷贝的数据。

len - 数据长度。

➤ Output argument 输出函数

positive - copied data's length.

negative - operation failure

正数 - 拷贝的数据的长度。

负数 - 操作失败。

4.2.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2.3 Copy data from PM_BUS buffer

4.2.3.1 Function Description 功能描述

This function is used to get data from PM_BUS buffer.

本函数用于从 PM_BUS 缓存获取数据。

4.2.3.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t copy_from_pmbuf(int8_t bus_no, int8_t buf_no, uint8_t *buf)

➤ Input argument 输入参数

bus_no - PM_BUS' bus no.

buf_no - PM_BUS' buffer no.

buf - data buffer .

bus_no - PM_BUS 的总线号。

buf_no - PM_BUS 的缓存号。

buf - 数据缓存。

➤ Output argument 输出函数

positive - copied data's length.

negative - operation failure

正数 - 拷贝的数据的长度。

负数 - 操作失败。

4.2.3.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2.4 Start PM_BUS transmit

4.2.4.1 Function Description 功能描述

This function starts PM_BUS transmit.

该函数发起 PM_BUS 传输。

4.2.4.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t enable_pm_send(int8_t bus_no,int8_t buf_no)

➤ Input argument 输入参数

bus_no - PM_BUS' bus no.

buf_no - PM_BUS' buffer no.

bus_no - PM_BUS 的总线号。

buf_no - PM_BUS 的缓存号。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.2.4.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2.5 Check PM_BUS receive status

4.2.5.1 Function Description 功能描述

This function is used to check the status of PM_BUS receive.

本函数用于检查 PM_BUS 接收的状态。

4.2.5.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t check_pm_recv(int8_t bus_no,int8_t buf_no)

➤ Input argument 输入参数

bus_no - PM_BUS' bus no.

buf_no - PM_BUS' buffer no.

bus_no - PM_BUS 的总线号。

buf_no - PM_BUS 的缓存号。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.2.5.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.2.6 Check PM_BUS send status

4.2.6.1 Function Description 功能描述

This function is used to check the status of PM_BUS send.

本函数用于检查 PM_BUS 发送的状态。

4.2.6.2 Argument Description 参数说明

➤ Function Definition 函数定义

int16_t check_pm_send(int8_t bus_no)

➤ Input argument 输入参数

bus_no - PM_BUS' bus no.

bus_no - PM_BUS 的总线号。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.2.6.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.3 IP_BUS driver IP_BUS 驱动

SWDD-PM-DD_NSafR_NSecR_A_003

This sub-function is used to transmit data over the IP_BUS.

本子功能用于在 IP_BUS 上传输数据。

4.3.1 IP_BUS send data

4.3.1.1 Function Description 功能描述

This function is used to send data over IP_BUS.

本函数用于在 IP_BUS 上发送数据。

4.3.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT16 ipbus_write(UINT8 opt,UINT8 *data,UINT32 offset,UINT32 len)

➤ Input argument 输入参数

opt - send option.

data - data to send

offset -address of the data send to.

len - data length.

opt - 发送选项。

data - 要发送的数据。

offset - 数据发送到的地址。

len - 数据长度。

➤ Output argument 输出函数

positive - sended data's length.

negative - operation failure

正数 - 发送的数据的长度。

负数 - 操作失败。

4.3.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.3.2 IP_BUS receive data

4.3.2.1 Function Description 功能描述

This function is used to receive data from IP_BUS.

本函数用于从 IP_BUS 接收数据。

4.3.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT16 ipbus_read(UINT8 opt, UINT8 *data, UINT32 offset, UINT32 len)

➤ Input argument 输入参数

opt - receive option.

data - store received data

offset - address of the data received from.

len - data length.

opt - 接收选项。

data - 保存接收到的数据。

offset -接收到的数据的地址。

len - 数据长度。

➤ Output argument 输出函数

positive - sended data's length.

negative - operation failure

正数 - 发送的数据的长度。

负数 - 操作失败。

4.3.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.4 PCIe driver PCIe 驱动

SWDD-PM-DD_NSafR_NSecR_A_004

This sub-function is used to transmit data over PCIe.

本子功能用于周期通过 PCIe 传输数据。

4.4.1 PCIe send data

4.4.1.1 Function Description 功能描述

This function is used to send data over PCIe.

本函数用于通过 PCIe 发送数据。

4.4.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
void pcie_write(uint8_t *dst, uint8_t *src, uint32_t len)
```

➤ Input argument 输入参数

dst - destination offset address in PCIe region.

src - source address in memory.

len - data length.

dst - PCIe 域中的目标偏移地址。

src - 内存上的源地址。

len - 数据长度。

➤ Output argument 输出函数

Length of the sended data.

发出数据的长度。

4.4.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.4.2 PCIe receive data

4.4.2.1 Function Description 功能描述

This function is used to receive data from PCIe.

本函数用于通过 PCIe 接收数据。

4.4.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
void pcie_read(uint8_t *dst, uint8_t *src, uint32_t len)
```

➤ Input argument 输入参数

dst - destination address in memory.

src - source offset address in PCIe region.

len - data length.

dst -内存上的目标地址。

src -PCIe 域中的源偏移地址。

len - 数据长度。

➤ Output argument 输出函数

Length of the received data.

接收到的数据的长度。

4.4.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.5 SPI driver SPI 驱动

SWDD-PM-DD_NSafR_NSecR_A_005

This sub-function is used to transmit data via SPI.

本子功能用于周期通过 SPI 传输数据。

4.5.1 Read SPI

4.5.1.1 Function Description 功能描述

This function is used to get data via SPI.

本函数用于通过 SPI 获取数据。

4.5.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
INT32 sysSpiReadData(UINT32 datalen, UINT8 * data);
```

➤ Input argument 输入参数

datalen - data length.

data - data buffer.

datalen - 数据长度。

data - 数据缓存。

➤ Output argument 输出函数

0 - operation succeeds.

-1 - operation failure.

0 - 操作成功。

-1 - 操作失败。

4.5.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.5.2 Write SPI

4.5.2.1 Function Description 功能描述

This function is used to send data via SPI.

本函数用于通过 SPI 发送数据。

4.5.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
INT32 sysSpiWriteData(UINT32 cs,UINT32 cmdlen,UINT32 datalen,UINT8 * cmd,UINT8 *  
data,UINT32 mode);
```

➤ Input argument 输入参数

cs - SPI chip select.

cmdlen - command length.

datalen - data length.

data - data to write.

mode - SPI work mode.

cs - SPI 片选。

cmdlen - 命令长度。

datalen - 数据长度。

data - 要写的数据。

mode - SPI 工作模式。

➤ Output argument 输出函数

0 - operation succeeds.

0 - 操作成功.

4.5.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.6 Nor Flash driver Nor Flash 驱动

SWDD-PM-DD_NSafR_NSecR_A_006

This sub-function is used to access data on NOR Flash.

本子功能用于访问 NOR Flash 上的数据。

4.6.1 Eraase NOR Flash

4.6.1.1 Function Description 功能描述

This function is used to erase NOR Flash.

本函数用于擦除 NOR Flash。

4.6.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

int lld_erase(int no, int num)

➤ Input argument 输入参数

no - erase from this block no.

num - block number to be erase.

no - 从该块号开始擦除。

num - 要擦除的块数。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.6.1.3 处理流程

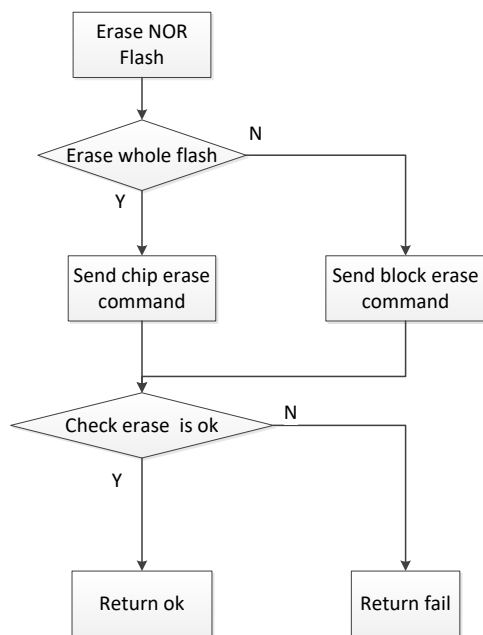


Figure 4-1 the process of write NOR Flash

图 4-1 写 NOR Flash 的过程

4.6.2 Write NOR Flash

4.6.2.1 Function Description 功能描述

This function is used write data to NOR Flash.

本函数用于将数据写入 NOR Flash。

4.6.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

int lld_write (ADDRESS addr, uchar *src, unsigned int cnt)

➤ Input argument 输入参数

addr - data is write to this address.

src - data address.

cnt - data length.

addr - 数据要写到这个地址。

src - 数据地址。

cnt - 数据长度。

➤ **Output argument 输出函数**

The length of the data is actually written.

实际写入的数据的长度。

4.6.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.6.3 Read NOR Flash

4.6.3.1 Function Description 功能描述

This function is used to read data from NOR Flash.

本函数用于从 Flash 读出数据。

4.6.3.2 Argument Description 参数说明

➤ **Function Definition 函数定义**

int lld_read (ADDRESS addr , uchar* buf, unsigned int cnt)

➤ **Input argument 输入参数**

addr - data is read from this address.

src - data address.

cnt - data length.

addr - 数据要从这个地址读出。

src - 数据地址。

cnt - 数据长度。

➤ **Output argument 输出函数**

The length of the data actually read.

实际读出的数据的长度。

4.6.3.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.7 SPI Flash driver SPI Flash 驱动

SWDD-PM-DD_NSafR_NSecR_A_007

This sub-function is used to access data on SPI Flash.

本子功能用于访问 SPI Flash 上的数据。

4.7.1 Read data from SPI Flash

4.7.1.1 Function Description 功能描述

This function is used to read data from SPI Flash.

本函数用于从 SPI Flash 读取数据。

4.7.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT32 spiflash_read(UINT32 addr, UINT8 *buf, UINT32 cnt)

➤ Input argument 输入参数

addr - Data address in flash.

buf - Data buffer.

cnt - Data length.

addr - 数据在 Flash 上的地址。

buf - 数据缓存。

cnt - 数据长度。

➤ Output argument 输出函数

negative - operation failure.

The length of the data actually read.

负数 - 操作失败。

实际读出的数据的长度。

4.7.1.3 处理流程

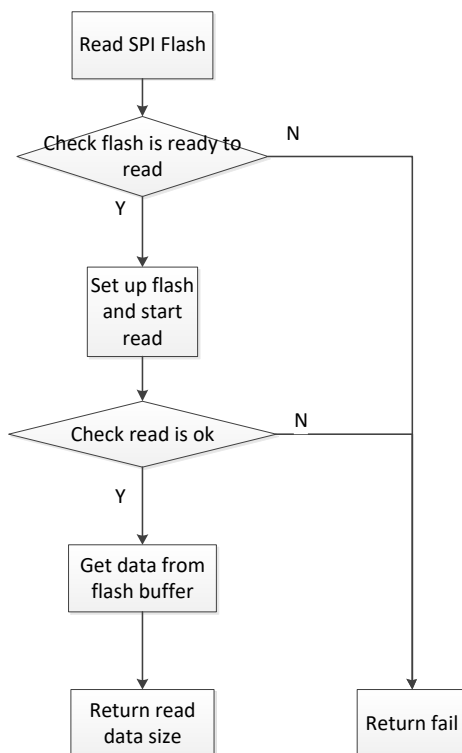


Figure 4-2 the process of read SPI Flash

图 4-2 读 SPI Flash 的过程

4.7.2 Write data to SPI Flash

4.7.2.1 Function Description 功能描述

This function is used write data to SPI Flash.

本函数用于把数据写到 SPI Flash。

4.7.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT32 spiflash_write(UINT32 addr, UINT8 *buf, UINT32 cnt)

➤ Input argument 输入参数

addr - data address in flash.

buf - source data buffer.

cnt - data length.

addr - 数据在 Flash 上的地址。

buf - 源数据缓存。

cnt - 数据长度。

➤ Output argument 输出函数

negative - operation failure.

The length of the data is actually written.

负数 - 操作失败。

实际写入的数据的长度。

4.7.2.3 处理流程

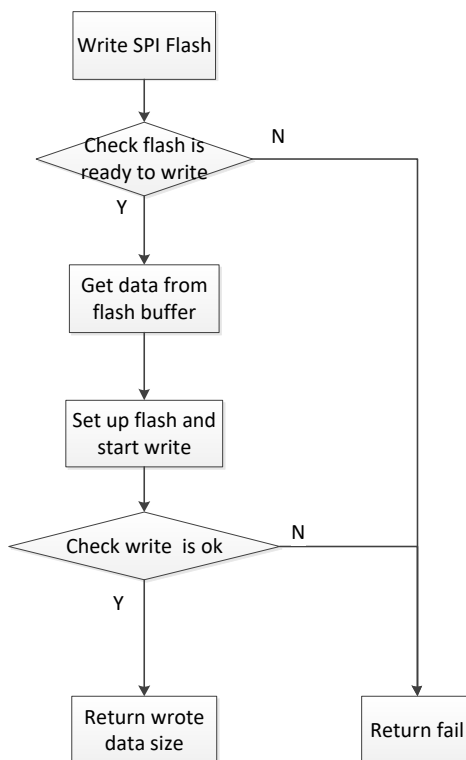


Figure 4-3 the process of write SPI Flash

图 4-3 写 SPI Flash 的过程

4.7.3 Erase SPI Flash

4.7.3.1 Function Description 功能描述

This function is used to erase SPI Flash.

本函数用于擦除 SPI Flash。

4.7.3.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT32 spiflash_erase(INT32 no , INT32 num)

➤ Input argument 输入参数

no - erase from this block no.

num - block number to be erase.

no - 从该块号开始擦除。

num - 要擦除的块数。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.7.3.3 处理流程

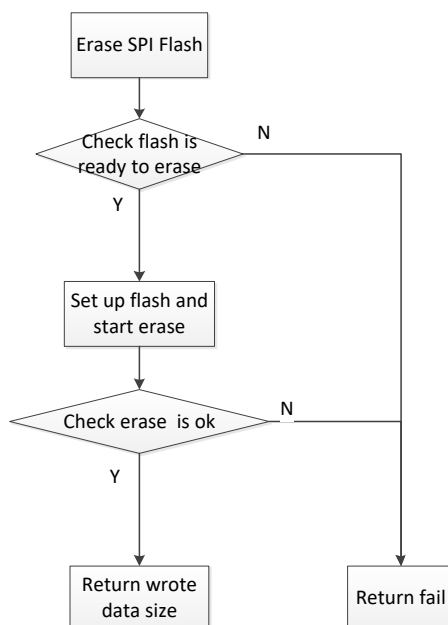


Figure 4-4 the process of erase SPI Flash

图 4-4 擦除 SPI Flash 的过程

4.8 FPGA driver FPGA 驱动

SWDD-PM-DD_NSafR_NSecR_A_008

This sub-function is used to process registers on FPGA.

本子功能用于处理 FPGA 上的寄存器。

4.8.1 Read FPGA

4.8.1.1 Function Description 功能描述

This function is used to read FPGA register.

本函数用于读取 FPGA 寄存器。

4.8.1.2 Argument Description 参数说明

- Function Definition 函数定义

uint16_t fpga_read(uint32_t reg_num)

- Input argument 输入参数

reg_num - register number.

reg_num - 寄存器号。

- Output argument 输出函数

Register's value.

寄存器的值。

4.8.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.8.2 Write FPGA

4.8.2.1 Function Description 功能描述

This function is used to modify FPGA register.

本函数用于修改 FPGA 寄存器。

4.8.2.2 Argument Description 参数说明

- Function Definition 函数定义

void fpga_write(uint32_t reg_num, uint16_t value)

- Input argument 输入参数

reg_num - register number.

value - new vlue of register.

reg_num - 寄存器号。

value - 寄存器的新值。

- Output argument 输出函数

No.

无。

4.8.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.9 DFS driver DFS 驱动

SWDD-PM-DD_NSafR_NSecR_A_009

This sub-function as an abstraction layer, and it is used to isolate the upper application from the operation of the peripheral devices.

本子功能作为抽象层，用于将对外设的操作与上层应用隔离开。

4.9.1 DFS Read

4.9.1.1 Function Description 功能描述

This function is an abstract interface of read device.

本函数是读外设的抽象接口。

4.9.1.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT32 dfs_read(INT32 dfd, UINT32 src, UINT8 *buf,UINT32 cnt)

➤ Input argument 输入参数

dfd - device ID in DFS.

src - source data address.

buf - read data buffer.

cnt - data length.

dfd - 设备在 DFS 中的 ID。

src - 数据源地址。

buf - 读数据缓存。

cnt - 数据长度

➤ Output argument 输出函数

negative - operation failure.

Other value is the length of the data actually read.

负数 - 操作失败。

其它值为实际读出的数据的长度。

4.9.1.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.9.2 DFS Write

4.9.2.1 Function Description 功能描述

This function is an abstract interface of write device.

本函数是写外设的抽象接口。

4.9.2.2 Argument Description 参数说明

➤ Function Definition 函数定义

INT32 dfs_write(INT32 dfd, UINT32 dst,UINT8 *src,UINT32 cnt)

➤ Input argument 输入参数

dfd - device ID in DFS.

dst - destination data address.

src - write data buffer.

cnt - data length.

dfd - 设备在 DFS 中的 ID。

src - 数据目标地址。

buf - 写数据缓存。

cnt - 数据长度。

➤ Output argument 输出函数

negative - operation failure.

Other value is the length of the data is actually written.

负数 - 操作失败。

其它值为实际写入的数据的长度。

4.9.2.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.9.3 DFS Erase

4.9.3.1 Function Description 功能描述

This function is an abstract interface of erase device.

本函数是擦除外设的抽象接口。

4.9.3.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
INT32 dfs_erase(INT32 dfd,UINT32 no,UINT32 num)
```

➤ Input argument 输入参数

dfd - device ID in DFS.

no - erase from this block no.

num - block number to be erase.

dfd - 设备在 DFS 中的 ID。

no - 从该块号开始擦除。

num - 要擦除的块数。

➤ Output argument 输出函数

0 - operation succeeds.

negative - operation failure

0 - 操作成功。

负数 - 操作失败。

4.9.3.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.9.4 DFS Initialize

4.9.4.1 Function Description 功能描述

This function is used to initialize DFS structure.

本函数用于初始化 DFS 结构体。

4.9.4.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
void dfs_init(void)
```


- Input argument 输入参数

No.

无。

- Output argument 输出函数

No.

无。

4.9.4.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.9.5 Register device to DFS

4.9.5.1 Function Description 功能描述

This function is used register a device to DFS.

本函数用于向 DFS 注册一个设备。

4.9.5.2 Argument Description 参数说明

- Function Definition 函数定义

INT32 dfs_register(struct dfs_ops *dops)

- Input argument 输入参数

dops - a structure used to descript device.

dops - 描述外设的结构体。

- Output argument 输出函数

0 - operation succeeds.

Other value is the DFS ID of a device.

0 - 操作成功。

其它值为设备的 DFS ID。

4.9.5.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

4.9.6 Deregister device from DFS

4.9.6.1 Function Description 功能描述

This function is used to deregister a device from DFS.

本函数用于从 DFS 卸载一个设备。

4.9.6.2 Argument Description 参数说明

➤ Function Definition 函数定义

```
void dfs_unregister(INT32 dfd)
```

➤ Input argument 输入参数

dfd - device ID in DFS.

dfd - 设备在 DFS 中的 ID。

➤ Output argument 输出函数

No.

无。

4.9.6.3 处理流程

This function has no branch and the processing flow is omitted.

此函数无分支，流程图省略。

——以下无正文