



XR806 FDCM 中间件 开发指南

版本号：1.0

发布时间：2020-11-17

版本历史

版本	日期	责任人	版本描述
1.0	2020-11-17	AWA 1680	创建文档。

目录

版本历史.....	i
目录.....	ii
表格目录.....	iv
1 前言.....	1
1.1 文档简介.....	1
1.2 目标读者.....	1
1.3 适用范围.....	1
1.4 文档约定.....	1
1.4.1 标志说明.....	1
1.4.2 地址与数据描述方法约定.....	1
1.4.3 数值单位约定.....	2
2 概述.....	3
2.1 背景说明.....	3
2.2 规格特性.....	3
2.3 文件位置.....	3
3 技术说明.....	4
3.1 模块结构.....	4
3.2 FDCM 管理的 Flash 区域.....	4
3.3 FDCM 读写数据原理.....	5
4 应用说明.....	6
4.1 应用简述.....	6
4.2 配置说明.....	6
4.3 接口说明.....	7
4.3.1 开关接口.....	7
4.3.1.1 fdcm_open.....	7
4.3.1.2 fdcm_close.....	8
4.3.2 读写擦接口.....	9
4.3.2.1 fdcm_read.....	9
4.3.2.2 fdcm_write.....	9
4.3.2.3 fdcm_erase.....	10
4.4 注意事项.....	10
5 示例说明.....	11

5.1 示例简介.....	11
5.1.1 获取方式.....	11
5.1.2 准备工作.....	11
5.1.3 操作步骤.....	11
5.2 关键实现.....	11
5.3 效果展示.....	13
6 常见问题.....	14
6.1 FDCM 打开失败.....	14
6.2 FDCM 写入失败.....	14
6.3 FDCM 读取失败.....	14
附录 A：术语表.....	15

表格目录

表 2-1	FDCM 中间件模块功能特性.....	3
表 2-2	FDCM 中间件的文件位置.....	3
表 4-1	XR806 FDCM 模块接口配置说明.....	6
表 4-2	FDCM 中间件模块接口简介.....	7
表 4-3	fdcm_open 接口函数说明.....	7
表 4-4	fdcm_handle_t 结构体的成员说明.....	8
表 4-5	fdcm_close 接口函数说明.....	8
表 4-6	fdcm_read 接口函数说明.....	9
表 4-7	fdcm_write 接口函数说明.....	9
表 4-8	fdcm_erase 接口函数说明.....	10
表 6-1	FDCM 打开失败原因分析.....	14
表 6-2	FDCM 写入失败原因分析.....	14
表 6-3	FDCM 读取失败原因分析.....	14
表 A-1	术语表.....	15

1 前言

1.1 文档简介

本文档主要介绍 FDCM 模块的基本原理和使用方法，并说明使用该模块的注意事项，以方便开发者正确使用该模块进行开发。

1.2 目标读者

XR806 SDK 用户、FDCM 开发使用人员。

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

符号	例子	说明
Ox	0x0200, 0x79	地址或数据以 16 进制表示。
Ob	0b010, 0b00 000 111	数据采用二进制表示(寄存器描述除外)。
X	00X, XX1	数据描述中，X 代表 0 或 1。 例如，00X 代表 000 或 001；XX1 代表 001, 011, 101 或 111。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

类型	符号	对应数值
数据容量（如 NAND 容量）	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率，数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

2 概述

2.1 背景说明

FDCM（Flash Data Chunk Management）模块用于在指定的 Flash 区域对数据块进行管理。使用者调用该模块接口可方便地将数据写到 Flash 指定区域，并在需要使用数据时从 Flash 中读出。

FDCM 模块对数据块管理的主要特点为：对同一区域多次写入数据块后，从该区域读出的总是最后一次写入的数据块内容。适合用户在保存 SSID（Service Set Identifier），PSK（pre-shared key），以及其他的一些用户信息时使用，可以避免频繁地对 Flash 进行擦除操作，延长 Flash 使用寿命。

2.2 规格特性

FDCM 中间件模块提供了以下功能特性。

表 2-1 FDCM 中间件模块功能特性

规格类型	规格功能特性描述	备注
基本功能	支持对 Flash 区域读、写、擦除等操作；支持对同一区域多次写入	
磨损均衡	支持 Flash 磨损保护	
对 Flash 要求	不限 Flash 型号与容量	

2.3 文件位置

以 SDK 包为根目录，本中间件涉及到的主要文件位置如下。

表 2-2 FDCM 中间件的文件位置

组件名	文件分类	文件位置
FDCM	源码文件	./src/image/fdcm.c
	头文件	./include/image/fdcm.h
	示例工程	./project/example/fdcm/



说明

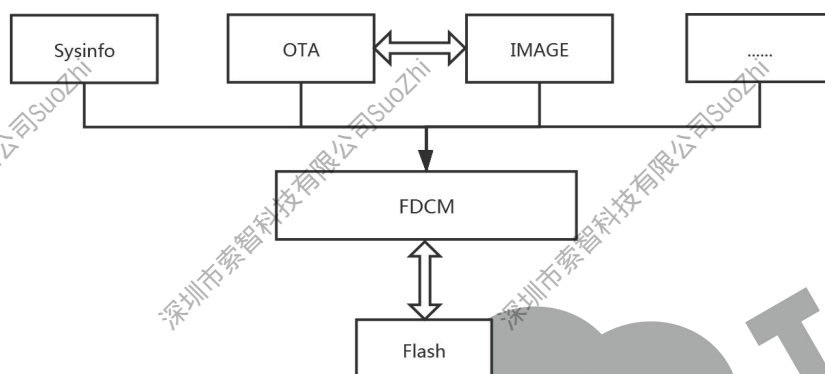
XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

3 技术说明

3.1 模块结构

FDCM 模块与 SDK 中其他模块之间的关系如下图所示。Sysinfo, OTA 和 IMAGE 等模块中都有使用到 FDCM 模块接口，保存其对应的数据。其中，OTA 和 IMAGE 模块还会根据写在 FDCM 中的数据进行一些判断和配置。

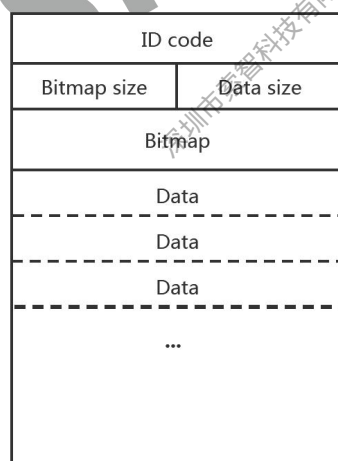
图 3-1 FDCM 与其他模块之间的关系



3.2 FDCM 管理的 Flash 区域

使用 FDCM 模块需要指定一段 Flash 区域来存储数据块，该区域必须与 Flash 可擦除的 Block 对齐。FDCM 模块管理的 Flash 区域内部结构如下图所示。

图 3-2 FDCM 管理的 Flash 区域内部结构示意图



图中这一段 Flash 区域内部可划分为三个部分：FDCM header 区域、Bitmap 区域和 Data 区域。

FDCM header 区域包含 ID code、Bitmap size 和 Data size。其中，ID code 是一个特定值，作为 FDCM 管理的一段 Flash 区域的标志。

Bitmap size 表示 Bitmap 区域的大小，在第一次写入数据块时根据数据块的大小和这一段 Flash 区域的大小计算得到。Bitmap 用于标记最新写入的数据块位于 Data 区域的具体位置。

Data size 表示 Data 区域中一个数据块的大小，在第一次写入数据块时确定，Data 区域中可存储多个大小相同的数据块。

3.3 FDCM 读写数据原理

第一次写入数据块时，FDCM 模块将构建上述内部结构。然后修改 Bitmap，并将数据块写在 Data 区域第一个数据块位置。之后每次写入相同大小数据块时，都修改一次 Bitmap，并将数据块写在前一次写入数据块的后一个位置。

当 Data 区域写满后，将擦除这一段 Flash 区域，并重复第一次写入数据块时的操作。当某次写入的数据块大小与上一次写入的数据块大小不同（即写入数据块大小不等于 Flash 区域上的 Data size）时，将擦除这一段 Flash 区域，并重复第一次写入数据块时的操作。

用 FDCM 读数据块时，将先核对 ID code，然后根据 Bitmap size、Data size 以及 Bitmap 找到并读取到最后一次写入的数据块。由于在第一次写数据块时才构建区域的内部结构，所以在未写过数据块的区域进行读数据块操作将报错。

4 应用说明

4.1 应用简述

FDCM 中间件模块已经内嵌到 XR806 SDK，应用步骤如下：

1. 确认/检查所用 Flash 设备是否能正常运行（Flash 模块设备配置可参考《XR806_Flash 驱动_开发指南》文档）。
2. 在用户应用代码中添加 FDCM 模块头文件（参见“2.3 文件位置”章节），调用 FDCM 接口进行使用（接口使用参见“4.3 接口说明”章节）。

4.2 配置说明

FDCM 模块无需进行特殊的工程配置，用户在自己代码中添加完头文件即可使用该模块对应的接口。在进行接口调用时，可以进行以下设置。

表 4-1 XR806 FDCM 模块接口配置说明

配置项	配置说明
Flash 设备号	设置说明： 此项配置为 Flash 次设备号的选择，若选用 image Flash，次设备号为 0，若选用其他 Flash，次设备号为用户配置。（详细可参见《XR806_Flash 驱动_开发指南》文档） 设置位置： 用户自己的应用 main.c 文件中（可参见“5 示例说明”章节） 设置方式： 可配置宏 FLASH_DEVICE_NUM 的值，如： #define FLASH_DEVICE_NUM 0 //表示使用 image Flash 涉及接口：fdcm_open()
FDCM 起始地址	设置说明： 此项配置为 FDCM 模块管理 Flash 区域的起始地址，该地址为 Flash 的内部物理地址。 设置位置： 用户自己的应用 main.c 文件中（可参见“5 示例说明”章节） 设置方式： 可配置宏 FDCM_FLASH_START_ADDR 的值，如设置起始地址为 1MB： #define FDCM_FLASH_START_ADDR (1024 * 1024)

配置项	配置说明
	涉及接口：fdcm_open()
FDCM 区域大小	设置说明： 此项配置为 FDCM 模块管理的 Flash 区域大小。 设置位置： 用户自己的应用 main.c 文件中（可参见“5 示例说明”章节） 设置方式： 配置宏 FDCM_SIZE 的值，如设置管理 Flash 区域大小为 4KB： #define FDCM_SIZE (4 * 1024) 涉及接口：fdcm_open()

4.3 接口说明

XR806 FDCM 中间件模块为用户提供 1 套 FDCM 操作接口，位于 SDK/include/image/fdcm.h 文件中，各接口的简要说明如下表所示。

表 4-2 FDCM 中间件模块接口简介

接口名	简要说明
开关接口	包括 2 个接口函数，用于 FDCM 操作区域的打开和关闭。
读写擦接口	包括 3 个接口函数，用于 FDCM 的读、写、擦除功能。

上述接口均于 SDK/src/image/fdcm.c 文件中实现，XR806 FDCM 中间件模块接口的详细说明如下。

4.3.1 开关接口

4.3.1.1 fdcm_open

表 4-3 fdcm_open 接口函数说明

信息项	说明
原型	fdcm_handle_t *fdcm_open(uint32_t flash, uint32_t addr, uint32_t size);
功能	创建 FDCM 操作区域。此区域必须与 Flash 可擦除的 block 大小对齐，即区域起始地址与 Flash 可擦除 block 对齐，且长度也为 Flash 可擦除 block 大小的整数倍。
参数	uint32_t flash 含义解释：Flash 设备号 使用说明：若选用 image Flash，设备号为 0，若选用其他 Flash，设备号为用户配置。（详细可参见《XR806 Flash 驱动_开发指南》文档）

信息项	说明
	uint32_t addr 含义解释：被管理的 Flash 区域的起始地址 使用说明：N/A uint32_t size 含义解释：被管理的 Flash 区域大小 使用说明：N/A
返回值	fdcm_handle_t *, 返回指定 Flash 区域进行读写操作的 handle, fdcm_handle_t 结构体成员可参见表 4-4 fdcm_handle_t 结构体的成员说明。

表 4-4 fdcm_handle_t 结构体的成员说明

成员项	说明
uint32_t flash;	含义解释：用于指定 Flash 设备的编号 使用说明：N/A
uint32_t addr;	含义解释：用于存放 Flash 区域的起始地址 使用说明：N/A
uint32_t size;	含义解释：用于存放 Flash 区域的大小 使用说明：N/A

4.3.1.2 fdcm_close

表 4-5 fdcm_close 接口函数说明

信息项	说明
原型	void fdcm_close(fdcm_handle_t *hdl);
功能	关闭 FDCM 操作区域，释放调用 fdcm_open()时创建的系统资源
参数	fdcm_handle_t *hdl 含义解释：FDCM 读写操作的 handle 使用说明：fdcm_handle_t 结构体成员可参见表 4-4 fdcm_handle_t 结构体的成员说明。
返回值	无

4.3.2 读写擦接口

4.3.2.1 fdc_read

表 4-6 fdc_read 接口函数说明

信息项	说明
原型	uint32_t fdc_read(fdc_handle_t *hdl, void *data, uint16_t data_size);
功能	读取 FDCM 区域数据块
参数	fdcm_handle_t *hdl 含义解释：FDCM 读写操作的 handle 使用说明：fdcm_handle_t 结构体成员可参见表 4-4 fdc_handle_t 结构体的成员说明。 void *data 含义解释：存放被读取出的数据的 buffer 使用说明：N/A uint16_t data_size 含义解释：待读取的数据块长度 使用说明：N/A
返回值	读取的数据块长度

4.3.2.2 fdc_write

表 4-7 fdc_write 接口函数说明

信息项	说明
原型	uint32_t fdc_write(fdc_handle_t *hdl, const void *data, uint16_t data_size);
功能	向 FDCM 区域写入数据块
参数	fdcm_handle_t *hdl 含义解释：FDCM 读写操作的 handle 使用说明：fdcm_handle_t 结构体成员可参见表 4-4 fdc_handle_t 结构体的成员说明。 const void *data 含义解释：待写入的数据块 使用说明：N/A uint16_t data_size 含义解释：待写入的数据块长度 使用说明：N/A
返回值	写入的数据块长度

4.3.2.3 fdc_m_erase

表 4-8 fdc_m_erase 接口函数说明

信息项	说明
原型	int fdc_m_erase(fdc_m_handle_t *hdl);
功能	擦除整个 FDCM 区域
参数	fdc_m_handle_t *hdl 含义解释：FDCM 读写操作的 handle 使用说明：fdcm_handle_t 结构体成员可参见表 4-4 fdc_m_handle_t 结构体的成员说明。
返回值	0：成功 -1：失败

4.4 注意事项

在使用 FDCM 模块时，有以下几点需要注意：

1. 用 FDCM 模块管理的 Flash 区域不要与其他模块（如 Image）使用的 Flash 区域产生冲突，避免相互影响。
2. 用 FDCM 模块管理的 Flash 区域必须与 Flash 可擦除的 block 对齐，即区域起始地址与 Flash 擦除 Block 的边缘对齐，并且区域大小为 Flash 可擦除最小 Block 大小的整数倍。
3. 先正确写入数据块后才能正确读出数据块，读数据块时的参数 data_size 应该与最后一次写数据块的参数 data_size 相同。
4. 连续多次写数据块允许每次写数据块的 data_size 不同，但根据 FDCM 读写数据原理，建议对同一 Flash 区域每次读写数据块的 data_size 尽可能一致，这样能减少对 Flash 擦除的次数。

5 示例说明

FDCM 接口用于在指定的 Flash 区域对数据块进行管理，下面以对 FDCM 区域的创建、读写、释放为例，简要说明 FDCM 接口的使用。

5.1 示例简介

FDCM 示例工程展示了 XR806 SDK 中使用 FDCM 保存数据的代码实现方法。

示例工程会在 Flash 中创建一个 FDCM 区域，用来存放信息。示例中会多次进行 FDCM 区域的写和读操作。通过串口打印，可以看出写进 FDCM 的信息和读取出的信息为一致。

5.1.1 获取方式

FDCM 示例有示例工程代码，位于 XR806 SDK 的/project/example/fdcm 目录，以下此示例工程简称为 FDCM 示例工程。



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

5.1.2 准备工作

FDCM 示例工程的硬件准备有如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

FDCM 示例工程的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》。

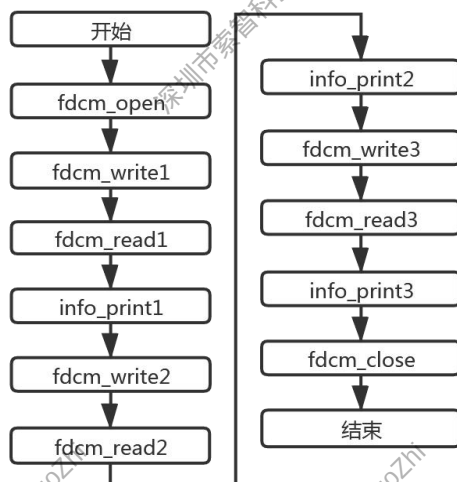
5.1.3 操作步骤

FDCM 示例工程无需控制命令，完成烧写后，复位即可，示例代码自动运行。

5.2 关键实现

系统平台的初始化在 XR806 SDK 的 platform_init()函数中完成，具体请阅读此函数。本文重点描述 FDCM 模块的打开/创建、写入 Wi-Fi ssid 和 password 信息、从 FDCM 区域读取信息、关闭/销毁 FDCM 模块的操作过程，示例代码流程如下。

图 5-1 FDCM 示例流程图



具体操作见下面的示例代码。

```

#define FLASH_DEVICE_NUM        0
#define FDCM_FLASH_START_ADDR  (1024 * 1024)
#define FDCM_SIZE                (4 * 1024)

struct wifi_info {
    char ssid[64];
    char password[64];
};
...
static int fdcm_example()
{
    fdcm_handle_t *fdcm;
    struct wifi_info wifi;
    struct wifi_info info;

    fdcm = fdcm_open(FLASH_DEVICE_NUM, FDCM_FLASH_START_ADDR, FDCM_SIZE);
    if (fdcm == NULL) {
        printf("fdcm open fail.\n");
        return -1;
    }

    printf("fdcm open success, flash addr:0x%x, flash size:%d\n", FDCM_FLASH_START_ADDR,
        FDCM_SIZE);
    printf("we can use fdcm to save info to flash.\n");

    printf("===fdcm write/read example1===\n");
    strcpy(wifi.ssid, "first_ssid");
    strcpy(wifi.password, "first_password");
    fdcm_write(fdcm, &wifi, sizeof(wifi));
    
```

```
fdcm_read(fdcm, &info, sizeof(info));
printf("[write ssid:%s, write password:%s] [read ssid:%s, read password:%s]\n", wifi.ssid,
       wifi.password, info.ssid, info.password);

printf("====fdcm write/read example2====\n");
strcpy(wifi.ssid, "second_ssid");
strcpy(wifi.password, "second_password");
fdcm_write(fdcm, &wifi, sizeof(wifi));
fdcm_read(fdcm, &info, sizeof(info));
printf("[write ssid:%s, write password:%s] [read ssid:%s, read password:%s]\n", wifi.ssid,
       wifi.password, info.ssid, info.password);

printf("====fdcm write/read example3====\n");
strcpy(wifi.ssid, "third_ssid");
strcpy(wifi.password, "third_password");
fdcm_write(fdcm, &wifi, sizeof(wifi));
fdcm_read(fdcm, &info, sizeof(info));
printf("[write ssid:%s, write password:%s] [read ssid:%s, read password:%s]\n", wifi.ssid,
       wifi.password, info.ssid, info.password);

fdcm_close(fdcm);

return 0;
}
```

5.3 效果展示

在评估板中运行 FDCM 示例程序后，在控制台中打印输出如下。

```
fdcm example start.
fdcm open success, flash addr:0x100000, flash size:4096
we can use fdcm to save info to flash.
====fdcm write/read example1====
[write ssid:first_ssid, write password:first_password] [read ssid:first_ssid, read password:first_password]
====fdcm write/read example2====
[write ssid:second_ssid, write password:second_password] [read ssid:second_ssid, read
password:second_password]
====fdcm write/read example3====
[write ssid:third_ssid, write password:third_password] [read ssid:third_ssid, read
password:third_password]
fdcm example end.
```

6 常见问题

6.1 FDCM 打开失败

表 6-1 FDCM 打开失败原因分析

可能原因	措施
Flash 地址参数没有与可擦除的 block 对齐	尝试传入 4k 对齐或者 64k 对齐的 flash 地址

6.2 FDCM 写入失败

表 6-2 FDCM 写入失败原因分析

可能原因	措施
没有先调用 fdc_open(), 句柄为空	请先调用 fdc_open(), 获取句柄后再进行写入操作
写入长度超出 open 时设定的 size 大小	检查写入长度和 open 时设定长度, 调整后再作尝试

6.3 FDCM 读取失败

表 6-3 FDCM 读取失败原因分析

可能原因	措施
该 flash 区域还没有写入过数据	先写入数据, 之后再进行读取操作
读取长度与保存的数据长度不匹配	确认之前保存的数据长度之后再尝试读取

附录 A：术语表

表 A-1 术语表

F		
FDCM	Flash Data Chunk Management	Flash 数据块管理
P		
PSK	pre-shared key	预共享密钥
S		
SSID	Service Set Identifier	服务集标识符

著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。