

系统控制

文件标识: RK-SYS1-MPI-MMZ

发布版本: V0.1.0

日期: 2021.5

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司 (“本公司”, 下同) 不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

MMZ模块实现内存管理等功能。

产品版本

芯片名称	内核版本
RK356X	4.19

读者对象

本文档 (本指南) 主要适用于以下工程师:

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V0.1.0	周弟东	2021-05-27	初始版本

目录

前言

目录

API 参考

- [RK_MPI_MMZ_Alloc](#)
- [RK_MPI_MMZ_Free](#)
- [RK_MPI_MMZ_Handle2PhysAddr](#)
- [RK_MPI_MMZ_Handle2VirAddr](#)
- [RK_MPI_MMZ_Handle2Fd](#)
- [RK_MPI_MMZ_GetSize](#)
- [RK_MPI_MMZ_Fd2Handle](#)
- [RK_MPI_MMZ_VirAddr2Handle](#)
- [RK_MPI_MMZ_PhyAddr2Handle](#)
- [RK_MPI_MMZ_IsCacheable](#)
- [RK_MPI_MMZ_FlushCacheStart](#)
- [RK_MPI_MMZ_FlushCacheEnd](#)
- [RK_MPI_MMZ_FlushCacheVaddrStart](#)
- [RK_MPI_MMZ_FlushCacheVaddrEnd](#)
- [RK_MPI_MMZ_FlushCachePaddrStart](#)
- [RK_MPI_MMZ_FlushCachePaddrEnd](#)

数据类型

- [RK_MMZ_ALLOC_TYPE_IOMMU](#)
- [RK_MMZ_ALLOC_TYPE_CMA](#)
- [RK_MMZ_ALLOC_CACHEABLE](#)
- [RK_MMZ_ALLOC_UNCACHEABLE](#)
- [RK_MMZ_SYNC_READONLY](#)
- [RK_MMZ_SYNC_WRITEONLY](#)
- [RK_MMZ_SYNC_RW](#)

API 参考

MMZ该功能模块为用户提供以下 API:

- [RK MPI MMZ Alloc](#): 申请用户缓存。
- [RK MPI MMZ Free](#): 释放用户缓存。
- [RK MPI MMZ Handle2PhysAddr](#): 获取用户缓存的物理地址。
- [RK MPI MMZ Handle2VirAddr](#): 获取用户缓存的虚地址。
- [RK MPI MMZ Handle2Fd](#): 获取用户缓存的fd。
- [RK MPI MMZ GetSize](#): 获取用户缓存的大小。
- [RK MPI MMZ Fd2Handle](#): 通过fd查找到对应的用户缓存。
- [RK MPI MMZ VirAddr2Handle](#): 通过虚地址查找对应的用户缓存。
- [RK MPI MMZ PhyAddr2Handle](#): 通过物理查找对应的用户缓存。

- [RK MPI MMZ IsCacheable](#)：查询用户缓存是否为cache 缓存。
- [RK MPI MMZ FlushCacheStart](#)：刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问前调用，当offset和length都等于0时候，执行full sync，否则执行partial sync。
- [RK MPI MMZ FlushCacheEnd](#)：刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问结束后调用，当offset和length都等于0时候，执行full sync，否则执行partial sync。
- [RK MPI MMZ FlushCacheVaddrStart](#)：刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问前调用，指定待刷新内存的虚拟地址及其长度，只支持partial sync。
- [RK MPI MMZ FlushCacheVaddrEnd](#)：刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问结束后调用，指定待刷新内存的虚拟地址及其长度，只支持partial sync。
- [RK MPI MMZ FlushCachePaddrStart](#)：刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问前调用，指定待刷新内存的物理地址及其长度，只支持partial sync。
- [RK MPI MMZ FlushCachePaddrEnd](#)：刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问结束后调用，指定待刷新内存的物理地址及其长度，只支持partial sync。

RK_MPI_MMZ_Alloc

【描述】
申请用户缓存。

【语法】

RK_S32 RK_MPI_MMZ_Alloc([MB_BLK](#) *pBlk, RK_U32 u32Length, RK_U32 u32Flags);

【参数】

参数名	描述	输入/输出
pBlk	缓存块指针	输出
u32Length	缓存块的大小	输入
u32Flags	内存标志，可同时或者分别带上是否支持Cache标记和物理内存是否连续标记，当前支持如下标志： RK_MMZ_ALLOC_CACHEABLE：支持Cache缓存 RK_MMZ_ALLOC_UNCACHEABLE：不支持Cache缓存 RK_MMZ_ALLOC_TYPE_IOMMU：带上此标志表示申请物理不连续内存 RK_MMZ_ALLOC_TYPE_CMA：带上此标志表示申请物理连续内存	输入

【返回值】

返回值	描述
0	成功
负值	失败

【注意】

- 无。

RK_MPI_MMZ_Free

【描述】
释放用户缓存。

【语法】

RK_S32 RK_MPI_MMZ_Free([MB_BLK](#) blk);

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入

【返回值】

返回值	描述
0	成功
负值	失败

【注意】

- 无。

RK_MPI_MMZ_Handle2PhysAddr

【描述】

获取用户缓存的物理地址。

【语法】

RK_U64 RK_MPI_MMZ_Handle2PhysAddr([MB_BLK](#) blk);

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入

【返回值】

返回值	描述
大于0	获取有效物理地址
0	失败

【注意】

- 无。

RK_MPI_MMZ_Handle2VirAddr

【描述】

获取用户缓存的虚地址。

【语法】

RK_VOID *RK_MPI_MMZ_Handle2VirAddr([MB_BLK](#) blk);

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入

【返回值】

返回值	描述
非RK_NULL	有效的虚拟地址
RK_NULL	获取虚拟地址失败

【注意】

- 无

RK_MPI_MMZ_Handle2Fd

【描述】

获取用户缓存的fd。

【语法】

RK_S32 RK_MPI_MMZ_Handle2Fd([MB_BLK](#) blk);

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入

【返回值】

返回值	描述
非负	有效的fd
负数	失败

【注意】

- 无

RK_MPI_MMZ_GetSize

【描述】

获取用户缓存的大小。

【语法】

RK_U64 RK_MPI_MMZ_GetSize([MB_BLK](#) blk);

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入

【返回值】

返回值	描述
非负	缓存块的大小
负数	失败

【注意】

- 无

RK_MPI_MMZ_Fd2Handle

【描述】

通过fd查找到对应的用户缓存。

【语法】

MB_BLK RK_MPI_MMZ_Fd2Handle(RK_S32 u32Fd);

【参数】

参数名	描述	输入/输出
u32Fd	fd值	输入

【返回值】

返回值	描述
非空	缓存块ID
RK_NULL	失败

【注意】

- 无

RK_MPI_MMZ_VirAddr2Handle

【描述】

通过虚拟地址查找对应的用户缓存。

【语法】

MB_BLK RK_MPI_MMZ_VirAddr2Handle(RK_VOID *pVirAddr);

【参数】

参数名	描述	输入/输出
pVirAddr	虚拟地址	输入

【返回值】

返回值	描述
非空	缓存块ID
RK_NULL	失败

RK_MPI_MMZ_PhyAddr2Handle

【描述】

通过物理地址查找对应的用户缓存。

【语法】

```
MB_BLK RK_MPI_MMZ_PhyAddr2Handle(RK_U64 u64phyAddr);
```

【参数】

参数名	描述	输入/输出
u64phyAddr	虚拟地址	输入

【返回值】

返回值	描述
非空	缓存块ID
RK_NULL	失败

RK_MPI_MMZ_IsCacheable

【描述】

查询用户缓存是否为cache 缓存。

【语法】

```
RK_S32 RK_MPI_MMZ_IsCacheable(MB\_BLK blk);
```

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入

【返回值】

返回值	描述
1	当前缓存是cache缓存
0	当前缓存非cache缓存
负值	查询失败

RK_MPI_MMZ_FlushCacheStart

【描述】

刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问前调用，当offset和length都等于0时候，执行full sync，否则执行partial sync。以[start](#)开始，以[end](#)结束， 其间是CPU对该内存的操作。

【语法】

```
RK_S32 RK_MPI_MMZ_FlushCacheStart(MB\_BLK blk, RK_U32 u32Offset, RK_U32 u32Length, RK_U32 u32Flags);
```

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入
u32Offset	指定刷新内存的偏移地址	输入
u32Length	需要刷新cache的长度	输入
u32Flags	读写标志，flag在start和end中需要保持一致： RK_MMZ_SYNC_READONLY：在start和end之间的代码，对指定内存cpu只做读操作 RK_MMZ_SYNC_WRITEONLY：在start和end之间的代码，对指定内存cpu只做写操作 RK_MMZ_SYNC_RW：在start和end之间的代码，对指定内存cpu即有读也有写	输入

【返回值】

返回值	描述
0	成功
负值	失败

RK_MPI_MMZ_FlushCacheEnd

【描述】

刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问结束后调用，当offset和length都等于0时候，执行full sync，否则执行partial sync。以[start](#)开始，以[end](#)结束， 其间是CPU对该内存的操作。

【语法】

```
RK_S32 RK_MPI_MMZ_FlushCacheEnd(MB\_BLK blk, RK_U32 u32Offset, RK_U32 u32Length, RK_U32 u32Flags);
```

【参数】

参数名	描述	输入/输出
blk	缓存块ID	输入
u32Offset	指定刷新内存的偏移地址	输入
u32Length	需要刷新内存的长度	输入
u32Flags	读写标志，flag在start和end中需要保持一致： RK_MMZ_SYNC_READONLY：在start和end之间的代码，对指定内存cpu只做读操作 RK_MMZ_SYNC_WRITEONLY：在start和end之间的代码，对指定内存cpu只做写操作 RK_MMZ_SYNC_RW：在start和end之间的代码，对指定内存cpu即有读也有写	输入

【返回值】

返回值	描述
0	成功
负值	失败

RK_MPI_MMZ_FlushCacheVaddrStart

【描述】

刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问前调用，指定待刷新内存的虚拟地址及其长度，只支持partial sync。以[start](#)开始，以[end](#)结束，其间是CPU对该内存的操作。

【语法】

```
RK_S32 RK_MPI_MMZ_FlushCacheVaddrStart(RK_VOID *pVirAddr, RK_U32 u32Length, RK_U32 u32Flags);
```

【参数】

参数名	描述	输入/输出
pVirAddr	指定刷新内存的虚拟地址	输入
u32Length	需要刷新内存的长度	输入
u32Flags	读写标志，flag在start和end中需要保持一致： RK_MMZ_SYNC_READONLY：在start和end之间的代码，对指定内存cpu只做读操作 RK_MMZ_SYNC_WRITEONLY：在start和end之间的代码，对指定内存cpu只做写操作 RK_MMZ_SYNC_RW：在start和end之间的代码，对指定内存cpu即有读也有写	输入

【返回值】

返回值	描述
0	成功
负值	失败

RK_MPI_MMZ_FlushCacheVaddrEnd

【描述】

刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问结束后调用，指定待刷新内存的虚拟地址及其长度，只支持partial sync。以[start](#)开始，以[end](#)结束，其间是CPU对该内存的操作。

【语法】

```
RK_S32 RK_MPI_MMZ_FlushCacheVaddrEnd(RK_VOID *pVirAddr, RK_U32 u32Length, RK_U32 u32Flags);
```

【参数】

参数名	描述	输入/输出
pVirAddr	指定刷新内存的虚拟地址	输入
u32Length	需要刷新内存的长度	输入
u32Flags	读写标志，flag在start和end中需要保持一致： RK_MMZ_SYNC_READONLY：在start和end之间的代码，对指定内存cpu只做读操作 RK_MMZ_SYNC_WRITEONLY：在start和end之间的代码，对指定内存cpu只做写操作 RK_MMZ_SYNC_RW：在start和end之间的代码，对指定内存cpu即有读也有写	输入

【返回值】

返回值	描述
0	成功
负值	失败

RK_MPI_MMZ_FlushCachePaddrStart

【描述】

刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问前调用，指定待刷新内存的物理地址及其长度，只支持partial sync。以[start](#)开始，以[end](#)结束，其间是CPU对该内存的操作。

【语法】

```
RK_S32 RK_MPI_MMZ_FlushCachePaddrStart(RK_U64 u64phyAddr, RK_U32 u32Length, RK_U32 u32Flags);
```

【参数】

参数名	描述	输入/输出
u64phyAddr	指定刷新内存的物理地址	输入
u32Length	需要刷新内存的长度	输入
u32Flags	读写标志，flag在start和end中需要保持一致： RK_MMZ_SYNC_READONLY：在start和end之间的代码，对指定内存cpu只做读操作 RK_MMZ_SYNC_WRITEONLY：在start和end之间的代码，对指定内存cpu只做写操作 RK_MMZ_SYNC_RW：在start和end之间的代码，对指定内存cpu即有读也有写	输入

【返回值】

返回值	描述
0	成功
负值	失败

RK_MPI_MMZ_FlushCachePaddrEnd

【描述】

刷新cache 里的内容到内存并且使 cache 里的内容无效，在cpu访问结束后调用，指定待刷新内存的物理地址及其长度，只支持partial sync。以[start](#)开始，以[end](#)结束， 其间是CPU对该内存的操作。

【语法】

```
RK_S32 RK_MPI_MMZ_FlushCachePaddrEnd(RK_U64 u64phyAddr, RK_U32 u32Length, RK_U32 u32Flags);
```

【参数】

参数名	描述	输入/输出
u64phyAddr	指定刷新内存的物理地址	输入
u32Length	需要刷新内存的长度	输入
u32Flags	读写标志，flag在start和end中需要保持一致： RK_MMZ_SYNC_READONLY：在start和end之间的代码，对指定内存cpu只做读操作 RK_MMZ_SYNC_WRITEONLY：在start和end之间的代码，对指定内存cpu只做写操作 RK_MMZ_SYNC_RW：在start和end之间的代码，对指定内存cpu即有读也有写	输入

【返回值】

返回值	描述
0	成功
负值	失败

数据类型

基本数据类型定义如下：

RK_MMZ_ALLOC_TYPE_IOMMU

【说明】

申请的物理内存为非连续物理地址内存。

【定义】

```
#define RK_MMZ_ALLOC_TYPE_IOMMU    0x00000000
```

RK_MMZ_ALLOC_TYPE_CMA

【说明】

申请的物理内存为连续物理地址内存。

【定义】

```
#define RK_MMZ_ALLOC_TYPE_CMA      0x00000001
```

RK_MMZ_ALLOC_CACHEABLE

【说明】

映射cached 属性的用户态虚拟地址，需要用户自己去刷新cache。

【定义】

```
#define RK_MMZ_ALLOC_CACHEABLE     0x00000000
```

RK_MMZ_ALLOC_UNCACHEABLE

【说明】

映射非cached 属性的用户态虚拟地址。

【定义】

```
#define RK_MMZ_ALLOC_UNCACHEABLE   0x00000010
```

RK_MMZ_SYNC_READONLY

【说明】

内存读操作。

【定义】

```
#define RK_MMZ_SYNC_READONLY       0x00000000
```

RK_MMZ_SYNC_WRITEONLY

【说明】

内存写操作。

【定义】

```
#define RK_MMZ_SYNC_WRITEONLY    0x00000001
```

RK_MMZ_SYNC_RW

【说明】

内存读和写操作。

【定义】

```
#define RK_MMZ_SYNC_RW          0x00000002
```