



Kip Broadhurst

摘要

本应用报告重点介绍了 Jacinto™ 7 防火墙的编程，用以将源自 A72/A53 的 DDR 事务与对已定义 DDR 存储器位置的非预期访问隔离开来。此外还包括关于在哪里可以找到防火墙相关文档的一般信息，以及对如何配置区域防火墙的理解。

可以从以下 URL 下载本文档所提到的工程配套资料和代码：<https://www.ti.com/cn/lit/zip/spracx6>。

内容

1 引言.....	2
2 防火墙文档.....	2
2.1 技术参考手册 (TRM).....	2
2.2 SDK TISCI 文档.....	2
2.3 SDK 防火墙文档.....	2
2.4 TI NDA 防火墙幻灯片组.....	2
3 防火墙定义和术语.....	3
4 SysConfig 工具.....	3
5 主防火墙与从防火墙.....	3
5.1 从防火墙.....	3
5.2 主防火墙.....	3
5.3 A72 主防火墙.....	4
6 防火墙的位置.....	5
6.1 示例.....	5
7 防火墙编程.....	8
7.1 示例 SBL 代码.....	8

插图清单

图 5-1. COMPUTE_CLUSTER0 概述.....	4
图 6-1. 示例代码.....	8

表格清单

表 6-1. 用于识别存储器范围的存储器区域.....	5
-----------------------------	---

商标

Jacinto™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

1 引言

通过管理对已定义的存储器范围的访问，防火墙可用于保护数据和器件配置。在考虑安全和防护解决方案时，隔离提供的防火墙功能非常有用。

Jacinto 7 系列器件均使用相同的方法来建立防火墙。本文档重点介绍了 J721E 器件和 Processor SDK 7.1 版本。就整个 Jacinto 7 系列和 Processor SDK 版本而言，从概念上讲，提供的示例在编程和代码位置方面应该相同。

文档中的示例代码和信息仅供参考，任何实现都需要对系统进行全面测试，而且需要理解对被测系统设置防火墙的原因。

2 防火墙文档

Jacinto 7 防火墙实现文档可在多个位置获得，该文档涉及 Jacinto 7 器件和 SDK S/W 架构上的许多不同组件。该文档参考并借鉴了可用的 TI 材料。

2.1 技术参考手册 (TRM)

被测器件的 TRM 有一个名为 *系统互连* 的小节。该节包含一个关于 *互连防火墙* 的子节。若要了解 Jacinto 7 防火墙解决方案，TI 建议用户首先参考该节。

ti.com 上提供了每个 TI 器件的 TRM。

2.2 SDK TISCI 文档

用户可以使用 TI 系统控制器接口 (TISCI) API 对防火墙进行编程。不允许直接在 J7 系列器件上对防火墙进行编程。

在开始使用 Jacinto 7 防火墙之前，建议先阅读有关防火墙的 TISCI 文档。该文档提供了关于术语的背景知识和总体概念。

- [防火墙 TISCI 说明](#)

用于防火墙配置的 TISCI 消息集很小。在设计如何配置系统时会面临复杂性挑战。TISCI 为防火墙提供的消息如下：

- 设置防火墙区域：
 - 指定防火墙 ID 和区域的设置、控制、权限和地址区域。
- 获取防火墙区域：
 - 查询指定防火墙 ID 和区域的当前配置
- 更改防火墙所有者：
 - 更改指定防火墙 ID 和区域的所有者。

2.3 SDK 防火墙文档

适用于 RTOS 的 TI Processor SDK 包含有关 SDK 的所有组件 (包括防火墙) 的文档。以下是 Processor SDK RTOS 7.1 版本的一些示例链接。对于被测 Jacinto 7 系列器件，每个页面都应该存在。下面提供的链接将指向 J721E 器件最新 SDK 版本。

- [J721E 防火墙说明](#)
- [J721E 主机说明](#)
- [防火墙常见问题解答](#)

2.4 TI NDA 防火墙幻灯片组

与德州仪器 (TI) 签订了 NDA 的客户可以使用一组演示防火墙设置的幻灯片。如需更多信息，请联系您当地的 TI 代表以获取 NDA 访问权限。

3 防火墙定义和术语

若要了解防火墙知识，在阅读 TI 文档时，还需要理解以下术语。防火墙常见问题解答是有助于加深理解的重要参考资料。

区域 一个已定义的存储器范围，防火墙权限和控制属性根据该范围进行存储。这些区域权限/属性用于过滤一个模块的互连事务。每个防火墙可以有 1 到 24 个区域。每个区域都包含以下寄存器：

- 控制
- 权限
- 起始/结束地址

主机 ID 主机 ID 是 SYSPFW 采用的软件概念，在 TISCI 中使用。主机 ID 表示处理实体。Jacinto 7 器件的主机 ID 在 TRM 和 SDK 文档中列出，也可以在头文件中查看。

特权 ID 特权 ID 是硬件级标识符。每台主机都对应一个特权 ID。一个特权 ID 可以代表一个或多个主机 ID。特权 ID 在 SDK 文档中列出，也可以在 SDK 头文件中查看。

防火墙 ID 用于唯一定义每个防火墙的标识符。

系统固件 “系统固件”是一个用于描述 TI 基础安全 (TIFS) 和资源管理 (RM)/电源管理 (PM) 服务的统称。

DMSC 安全管理器和设备管理器内核 (DMSC)。系统固件在 DMSC 上执行。

4 SysConfig 工具

TI 提供一款名为 **SysConfig** 的离线资源管理工具。**SysConfig** 可用于自动生成通用代码，然后可在 SDK 中用该代码对防火墙进行编程。防火墙的一般用途如下所列：

1. 使用 TI **SysConfig** 工具定义要编程的各种防火墙
2. **SysConfig** 工具将生成一个 .c 文件。
3. 将自动生成的 .c 文件中的内容集成到所选引导流程中

有关安装和使用 **SysConfig** 的说明，请参考嵌入的链接。

可以在本文档包含的示例代码中重复使用 **SysConfig** 的输出格式。

5 主防火墙与从防火墙

Jacinto7 互连上的所有模块和子系统可分为两类：主器件和从器件。主器件能够在系统中启动读写传输。另一方面，从器件依赖于主器件来执行在它们之间进行的传输。

5.1 从防火墙

文档中并未涉及这方面，但 **SysConfig** 工具的格式可在下面的示例代码中重复使用，因此添加从防火墙也只需一个简单步骤。从防火墙将在连接的从端（而不是主端）过滤事务。

5.2 主防火墙

对于主防火墙事务，事务在进入互连之前会被过滤。

看看 J721e 器件，它有三个防火墙：A72、C7x 和 DRU。本文档将详细介绍 A72 主防火墙。

5.3 A72 主防火墙

A72 本身是互连上的一个主器件，有一个能够过滤传出事务的主侧防火墙。图 5-1 (可在 DRA829/TDA4 器件的技术参考中找到) 显示了要编程的防火墙的位置。

A72 的主侧防火墙的防火墙 ID 为 257，如下面的代码示例所示，为 CSL_MSTR_FW_A72SS0_CORE0_CPU_0_CPU_0_MSMC_ID。

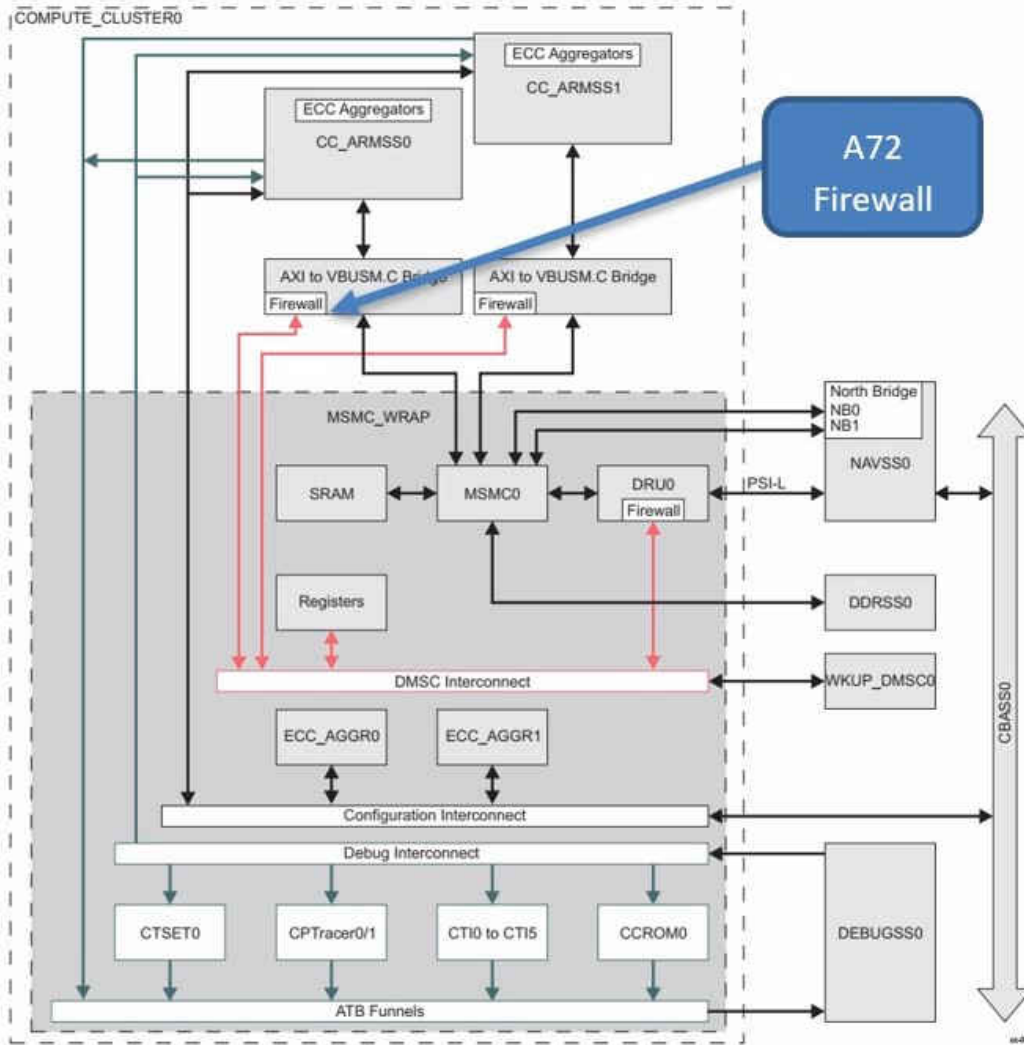


图 5-1. COMPUTE_CLUSTER0 概述

6 防火墙的位置

防火墙的 DDR 存储器范围因客户和用例而异。以默认的 TI SDK 版本为例，可以看到一些简单的初始步骤。

在为 vision_apps 解决方案构建 PSDK 的每个版本时，都会生成一个系统存储器映射。系统存储器映射显示了特定于每个内核的存储器，以及内核之间共享的存储器。在确定应针对 A72 访问在哪些区域设置防火墙时，推荐使用该存储器映射。

6.1 示例

在为系统进行定制时，应查看存储器映射以确定必须保护的任意区域，并查看应保护的存储器区域。如果 A72 不需要访问权限，则可以有选择性地对该存储器设置防火墙。

参考来自 Processor SDK QNX 7.1 的系统存储器映射，可以生成表 6-1。在该表中，可以查看存储器区域以识别对防火墙有意义的存储器范围，防止 A72 访问。

有了这些防火墙，所有 Vision Apps SDK 演示都将继续运行。如果 A72 软件无意中试图访问任何受防火墙保护的存储器区域，则会发生异常。

表 6-1. 用于识别存储器范围的存储器区域

名称	起始地址	结束地址	大小	属性	说明
L2RAM_C66x_1	0x00800000	0x00837FFF	224.00 KB	RWIX	L2 for C66x_1
L2RAM_C66x_2	0x00800000	0x00837FFF	224.00 KB	RWIX	L2 for C66x_2
MAIN_OCRAM_MCU2_0	0x03600000	0x0361FFFF	128.00 KB	RWIX	Main OCRAM for MCU2_0
MAIN_OCRAM_MCU2_1	0x03620000	0x0363FFFF	128.00 KB	RWIX	Main OCRAM for MCU2_1
L2RAM_C7x_1	0x64800000	0x64877FFF	480.00 KB	RWIX	L2 for C7x_1
L1RAM_C7x_1	0x64E00000	0x64E03FFF	16.00 KB	RWIX	L1 for C7x_1
MSMC_MPU1	0x70000000	0x7001FFFF	128.00 KB	RWIX	MSMC reserved for MPU1 for ATF
MSMC_C7x_1	0x70020000	0x707E7FFF	7.78 MB	RWIX	MSMC for C7x_1
MSMC_DMSC	0x707F0000	0x707FFFFFFF	64.00 KB	RWIX	MSMC reserved for DMSC IPC
DDR_MCU1_0_IPC	0xA0000000	0xA00FFFFFFF	1024.00 KB	RWIX	DDR for MCU1_0 for Linux IPC
DDR_MCU1_0_RESOURCE_TABLE	0xA0100000	0xA01003FF	1024 B	RWIX	DDR for MCU1_0 for Linux resource table
DDR_MCU1_0	0xA0100400	0xA0FFFFFFF	15.00 MB	RWIX	DDR for MCU1_0 for code/data
DDR_MCU2_0_IPC	0xA1000000	0xA10FFFFFFF	1024.00 KB	RWIX	DDR for MCU2_0 for Linux IPC
DDR_MCU2_0_RESOURCE_TABLE	0xA1100000	0xA11003FF	1024 B	RWIX	DDR for MCU2_0 for Linux resource table
DDR_MCU2_0	0xA1100400	0xA2FFFFFFF	31.00 MB	RWIX	DDR for MCU2_0 for code/data
DDR_MCU2_1_IPC	0xA3000000	0xA30FFFFFFF	1024.00 KB	RWIX	DDR for MCU2_1 for Linux IPC
DDR_MCU2_1_RESOURCE_TABLE	0xA3100000	0xA31003FF	1024 B	RWIX	DDR for MCU2_1 for Linux resource table
DDR_MCU2_1	0xA3100400	0xA4FFFFFFF	31.00 MB	RWIX	DDR for MCU2_1 for code/data
DDR_MCU3_0_IPC	0xA5000000	0xA50FFFFFFF	1024.00 KB	RWIX	DDR for MCU3_0 for Linux IPC
DDR_MCU3_0_RESOURCE_TABLE	0xA5100000	0xA51003FF	1024 B	RWIX	DDR for MCU3_0 for Linux resource table
DDR_MCU3_0	0xA5100400	0xA57FFFFFFF	7.00 MB	RWIX	DDR for MCU3_0 for code/data

表 6-1. 用于识别存储器范围的存储器区域 (continued)

名称	起始地址	结束地址	大小	属性	说明
DDR_MCU3_1_IPC	0xA5800000	0xA58FFFFFFF	1024.00 KB	RWIX	DDR for MCU3_1 for Linux IPC
DDR_MCU3_1_RESOURCE_TABLE	0xA5900000	0xA59003FF	1024 B	RWIX	DDR for MCU3_1 for Linux resource table
DDR_MCU3_1	0xA5900400	0xA5FFFFFFF	7.00 MB	RWIX	DDR for MCU3_1 for code/data
DDR_C66x_2_IPC	0xA6000000	0xA60FFFFFFF	1024.00 KB	RWIX	DDR for C66x_2 for Linux IPC
DDR_C66x_1_RESOURCE_TABLE	0xA6100000	0xA61003FF	1024 B	RWIX	DDR for C66x_1 for Linux resource table
DDR_C66x_1_BOOT	0xA6200000	0xA62003FF	1024 B	RWIX	DDR for C66x_1 for boot section
DDR_C66x_1	0xA6200400	0xA6FFFFFFF	14.00 MB	RWIX	DDR for C66x_1 for code/data
DDR_C66x_1_IPC	0xA7000000	0xA70FFFFFFF	1024.00 KB	RWIX	DDR for C66x_1 for Linux IPC
DDR_C66x_2_RESOURCE_TABLE	0xA7100000	0xA71003FF	1024 B	RWIX	DDR for C66x_2 for Linux resource table
DDR_C66x_2_BOOT	0xA7200000	0xA72003FF	1024 B	RWIX	DDR for C66x_2 for boot section
DDR_C66x_2	0xA7200400	0xA7FFFFFFF	14.00 MB	RWIX	DDR for C66x_2 for code/data
DDR_C7x_1_IPC	0xA8000000	0xA80FFFFFFF	1024.00 KB	RWIX	DDR for C7x_1 for Linux IPC
DDR_C7x_1_RESOURCE_TABLE	0xA8100000	0xA81003FF	1024 B	RWIX	DDR for C7x_1 for Linux resource table
DDR_C7x_1_BOOT	0xA8200000	0xA82003FF	1024 B	RWIX	DDR for C7x_1 for boot section
DDR_C7x_1_VECS	0xA8400000	0xA8403FFF	16.00 KB	RWIX	DDR for C7x_1 for vecs section
DDR_C7x_1_SECURE_VECS	0xA8600000	0xA8603FFF	16.00 KB	RWIX	DDR for C7x_1 for secure vecs section
DDR_C7x_1	0xA8604000	0xA8FFFFFFF	9.98 MB	RWIX	DDR for C7x_1 for code/data
IPC_VRING_MEM	0xAA000000	0xABFFFFFFF	32.00 MB		Memory for IPC Vring's.MUST be non-cached or cache-coherent
APP_LOG_MEM	0xAC000000	0xAC03FFFF	256.00 KB		Memory for remote core logging
TIOVX_OBJ_DESC_MEM	0xAC040000	0xADFDFFFF	31.62 MB		Memory for TI OpenVX shared memory.MUST be non-cached or cache-coherent
PCI_QUEUE_SHARED_MEM	0xADFE0000	0xADFEFFFF	64.00 KB		Memory for IPC over PCIe using shared memory.MUST be non-cached or cache-coherent
PCI_QUEUE_MIRROR_REMOTE_SHARED_MEM	0xADFF0000	0xADFFFFFFF	64.00 KB		Reserved Memory for RAT mapping of remote PCIe IPC shared memory.MUST be non-cached or cache-coherent

表 6-1. 用于识别存储器范围的存储器区域 (continued)

名称	起始地址	结束地址	大小	属性	说明
DDR_SHARED_MEM	0xAE000000	0xCDFFFFFFFF	512.00 MB		Memory for shared memory buffers in DDR
DDR_MCU2_0_NON_CACHE	0xCE000000	0xCE00FFFF	64.00 KB	RWIX	DDR for MCU2_0 for non-cached heap
DDR_MCU2_1_NON_CACHE	0xCE010000	0xD1FFFFFFF	63.94 MB	RWIX	DDR for MCU2_1 for non-cached heap
DDR_MCU1_0_LOCAL_HEAP	0xD2000000	0xD21FFFFFFF	2.00 MB	RWIX	DDR for MCU1_0 for local heap
DDR_MCU1_1_LOCAL_HEAP	0xD2200000	0xD23FFFFFFF	2.00 MB	RWIX	DDR for MCU1_1 for local heap
DDR_MCU2_0_LOCAL_HEAP	0xD2400000	0xD2BFFFFFFF	8.00 MB	RWIX	DDR for MCU2_0 for local heap
DDR_MCU2_1_LOCAL_HEAP	0xD2C00000	0xD3BFFFFFFF	16.00 MB	RWIX	DDR for MCU2_1 for local heap
DDR_MCU3_0_LOCAL_HEAP	0xD3C00000	0xD3DFFFFFFF	2.00 MB	RWIX	DDR for MCU3_0 for local heap
DDR_MCU3_1_LOCAL_HEAP	0xD3E00000	0xD3FFFFFFF	2.00 MB	RWIX	DDR for MCU3_1 for local heap
DDR_C66X_1_LOCAL_HEAP	0xD4000000	0xD4FFFFFFF	16.00 MB	RWIX	DDR for c66x_1 for local heap
DDR_C66X_1_SCRATCH	0xD5000000	0xD7FFFFFFF	48.00 MB	RWIX	DDR for c66x_1 for Scratch Memory
DDR_C66X_2_LOCAL_HEAP	0xD8000000	0xD8FFFFFFF	16.00 MB	RWIX	DDR for c66x_2 for local heap
DDR_C66X_2_SCRATCH	0xD9000000	0xDBFFFFFFF	48.00 MB	RWIX	DDR for c66x_2 for Scratch Memory
DDR_C7X_1_LOCAL_HEAP	0xDC000000	0xEBFFFFFFF	256.00 MB	RWIX	DDR for c7x_1 for local heap
DDR_C7X_1_SCRATCH	0xEC000000	0xF9FFFFFFF	224.00 MB	RWIX	DDR for c7x_1 for Scratch Memory
TIOVX_LOG_RT_MEM	0xFA000000	0xFAFFFFFFF	16.00 MB		Memory for TI OpenVX shared memory for Run-time logging. MUST be non-cached or cache-coherent
DDR_MCU1_1_IPC	0xFB000000	0xFB0FFFFFFF	1024.00 KB	RWIX	DDR for MCU1_1 for Linux IPC
DDR_MCU1_1_RESOURCE_TABLE	0xFB100000	0xFB1003FF	1024 B	RWIX	DDR for MCU1_1 for Linux resource table
DDR_MCU1_1	0xFB100400	0xFBFFFFFFF	15.00 MB	RWIX	DDR for MCU1_1 for code/data

使用该表并结合要设置防火墙保护的相邻存储器区域，以下 2 个 DDR 范围涵盖了 A72 不应访问的所有存储器位置。

起始地址	终止地址
0XA0000000	0XA8FFFFFFF
0xCE000000	0xFBFFFFFFF

若要防止 A72 访问这两个存储器范围，必须为每个存储器配置一个防火墙区域，并将区域权限指定为不允许任何 A72 访问。在以下示例代码中，A72 主防火墙的区域 0 设置为允许来自 A72 的所有访问，而区域 1 和区域 2 设置为防止 A72 访问需要设置防火墙保护的存储器范围。从 A72 的角度来看，可访问存储器的结果视图如下图所示。

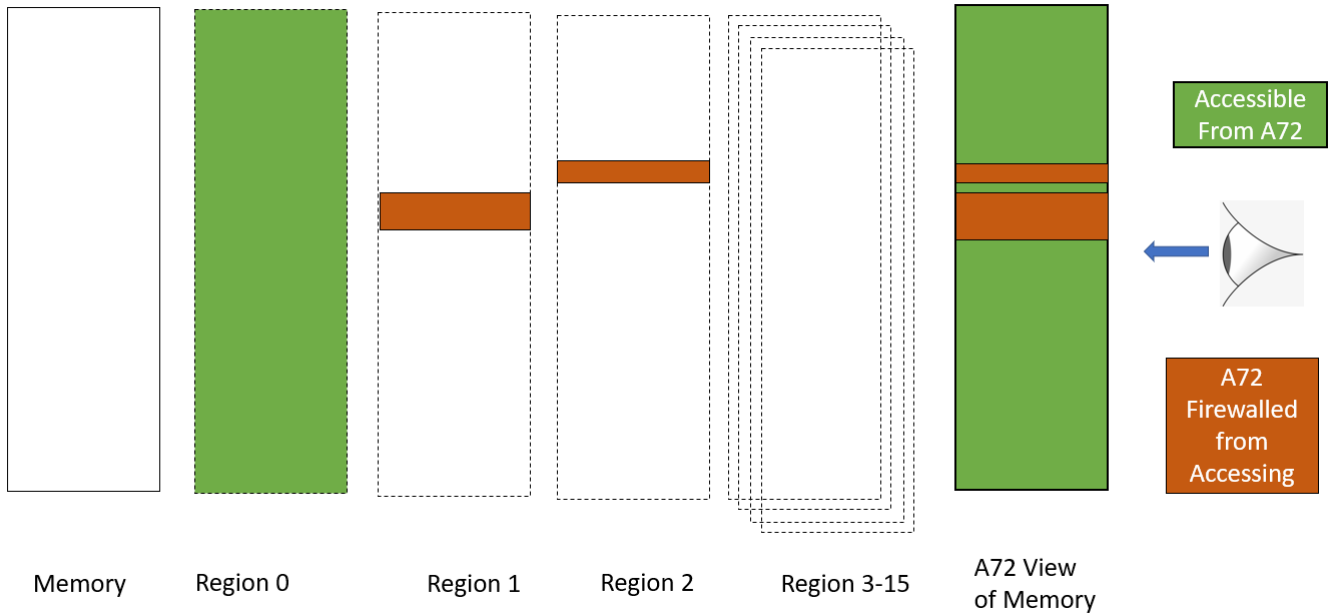


图 6-1. 示例代码

7 防火墙编程

请注意，防火墙寄存器的所有相关编程均通过 TISCI 完成。无法直接在 Jacinto 7 系列器件上对防火墙寄存器进行编程。

并非所有防火墙都是用户可编程的。对于可编程的防火墙，用户可以使用可用的 TI 文档或使用下面的示例代码。

Processor SDK 中有许多示例显示正在编程的防火墙，前面提到的 TI 文档中也有大量相关信息。

下面的示例代码旨在防止 A72 访问某些 DDR 存储器区域。将有许多不同的用例，允许或阻止对存储器位置或模块的访问。这些情况均可通过下面的示例框架来处理，只需扩展表条目即可。

7.1 示例 SBL 代码

添加防火墙时，任何事务都不能在传输过程中触碰防火墙。因此，建议在初始化期间添加防火墙。下面提供了一个选项，显示可添加到 SBL 引导流程以对 A72 主防火墙进行编程的函数。

如上所述，这个相同的框架可以重复用于从防火墙和其他主防火墙。只需要更新表条目。可以使用 [SysConfig](#) 工具生成表条目。

7.1.1 创建表

创建一个包含条目的表，其中每个条目代表一个防火墙区域。这种格式与 SysConfig 工具使用的 .c 输出格式相同。下面的示例为 A72 主防火墙创建了三个区域。

```

struct ti_sci_msg_fwl_region {
    uint16_t      fwl_id;
    uint16_t      region;
    uint32_t      n_permission_regs;
    uint32_t      control;
    uint32_t      permissions[FWL_MAX_PRIVID_SLOTS];
    uint64_t      start_address;
    uint64_t      end_address;
} __attribute__((packed));

void J721E_Set_Firewall(uint32_t isBuildHs)
{
    int32_t status = CSL_EFAIL;
    struct ti_sci_msg_fwl_region j721e_fwl_data[] = {

        /* compute_cluster Master firewall - background region 0 */
        {
            .fwl_id = CSL_MSTR_FW_A72SS0_CORE0_CPU_0_CPU_0_MSMC_ID,
            .region = 0,
            .n_permission_regs = 1,
            .control = 0x30A,
            .start_address = 0x00000000,
            .end_address = 0xFFFFFFFF,
            .permissions = { 0x1FFFF }, // PrivId 1U
        },
        /* compute_cluster Master firewall - region 1 */
        {
            .fwl_id = CSL_MSTR_FW_A72SS0_CORE0_CPU_0_CPU_0_MSMC_ID,
            .region = 1,
            .n_permission_regs = 1,
            .control = 0x20A,
            .start_address = 0xa0000000,
            .end_address = 0xa8ffffff,
            .permissions = { 0x10000 }, // PrivId 1U
        },
        /* compute_cluster Master firewall - region 2 */
        {
            .fwl_id = CSL_MSTR_FW_A72SS0_CORE0_CPU_0_CPU_0_MSMC_ID,
            .region = 2,
            .n_permission_regs = 1,
            .control = 0x20A,
            .start_address = 0xce000000,
            .end_address = 0xfbffffff,
            .permissions = { 0x10000 }, // PrivId 1U
        },
    };
};

```

- 区域 0 作为后台区域，使 A72 可以完全访问存储器范围。
- 然后区域 1 和 2 引入限制，确保 A72 对为这些区域定义的存储器范围没有任何权限。
- 请注意，对于所有三个区域，.permissions 条目上的 Privid 指示应将权限应用于哪个发起方。当位 [16] 设置为“1”时，这表明权限将应用于源自 A72 的事务。

7.1.2 解析防火墙区域表

对于防火墙区域表中的每个条目，

1. 使用定义的 TISCI API 来设置所有权
2. 使用定义的 TISCI API 对区域进行编程

在本例中，所有权设置为 MCU Boot Island (SBL 代码将在其中运行)。

```

uint32_t i = 0;
uint32_t j = 0;
uint32_t j721e_fwl_count = 3; // Number of entries

struct tisci_msg_fwl_set_firewall_region_resp respFwCtrl = {0};
struct tisci_msg_fwl_set_firewall_region_req reqFwCtrl;

for (i = 0; i < j721e_fwl_count; i++)
{
    /* Setting Owner */
    struct tisci_msg_fwl_change_owner_info_req req;
    req.fwl_id = (uint16_t)j721e_fwl_data[i].fwl_id;
    req.region = (uint16_t)j721e_fwl_data[i].region;
    req.owner_index = (uint8_t)HOST_ID_MCU_0_R5_1; // Cortex R5 context 1 on MCU island(Boot)

    struct tisci_msg_fwl_change_owner_info_resp resp = {0};          status =
    Sciclient_firewallChangeOwnerInfo(&req, &resp, SCICLIENT_SERVICE_WAIT_FOREVER);
    if (status != CSL_PASS)
    {
        SBL_log(SBL_LOG_ERR,"Firewall Unable to change Owner, %d\n", i);
        J721E_dump_owner_req(&req);
    }

    /* Setting Region */
    reqFwCtrl.fwl_id = (uint16_t)j721e_fwl_data[i].fwl_id;
    reqFwCtrl.region = (uint16_t)j721e_fwl_data[i].region;
    reqFwCtrl.n_permission_regs = (uint32_t)j721e_fwl_data[i].n_permission_regs;
    reqFwCtrl.control = (uint32_t)j721e_fwl_data[i].control;
    for(j = 0; j < reqFwCtrl.n_permission_regs; j++)
    {
        reqFwCtrl.permissions[j] = (uint32_t)j721e_fwl_data[i].permissions[j];
    }
    reqFwCtrl.start_address = j721e_fwl_data[i].start_address;
    reqFwCtrl.end_address = j721e_fwl_data[i].end_address;

    status = Sciclient_firewallSetRegion(&reqFwCtrl, &respFwCtrl,
    SCICLIENT_SERVICE_WAIT_FOREVER);
    if (status != CSL_PASS)
    {
        SBL_log(SBL_LOG_ERR,"Firewall entry/%d, region set failed.\n", i);
        J721E_dump_region_req(&reqFwCtrl);
    }
}
    
```

7.1.3 实用功能

新代码并不总是在第一次尝试时就能成功。下面的实用程序函数可以转储通过 SCI 发送的请求，从而在控制台端口上提供一些可读的输出。

```
void J721E_dump_owner_req(struct tisci_msg_fwl_change_owner_info_req *req)
{
    uint32_t i = 0;

    SBL_log(SBL_LOG_ERR, "\n");
    SBL_log(SBL_LOG_ERR, "Ownership Request:\n");
    SBL_log(SBL_LOG_ERR, "req.fwl_id           = 0x%x\n", req->fwl_id);
    SBL_log(SBL_LOG_ERR, "req.owner_index      = 0x%x\n", req->owner_index);
    SBL_log(SBL_LOG_ERR, "req.region           = 0x%x\n", req->region);

    uint8_t *bPtr = (uint8_t *) req;
    for (i = 0; (i < sizeof(struct tisci_msg_fwl_change_owner_info_req)); i++)
    {
        SBL_log(SBL_LOG_ERR, "%02x ", bPtr[i]);
    }
    SBL_log(SBL_LOG_ERR, "\n");
    return;
}

void J721E_dump_region_req(struct tisci_msg_fwl_set_firewall_region_req *req)
{
    uint32_t i = 0;
    uint32_t *startAddr;
    uint32_t *endAddr;

    startAddr = (uint32_t *) &req->start_address;
    endAddr = (uint32_t *) &req->end_address;

    SBL_log(SBL_LOG_ERR, "\n");
    SBL_log(SBL_LOG_ERR, "Region Set Request:\n");
    SBL_log(SBL_LOG_ERR, "req.fwl_id           = %d\n", req->fwl_id);
    SBL_log(SBL_LOG_ERR, "req.region           = %d\n", req->region);
    SBL_log(SBL_LOG_ERR, "req.control          = 0x%x\n", req->control);
    SBL_log(SBL_LOG_ERR, "req.start address    = 0x%x%x\n", startAddr[1], startAddr[0]);
    SBL_log(SBL_LOG_ERR, "req.end address      = 0x%x%x\n", endAddr[1], endAddr[0]);
    SBL_log(SBL_LOG_ERR, "req.n_permission_regs = 0x%x\n", req->n_permission_regs);
    for(i = 0; i < req->n_permission_regs; i++)
    {
        SBL_log(SBL_LOG_ERR, "req.permissions[%d] = 0x%x\n", i, req->permissions[0]);
    }
    SBL_log(SBL_LOG_ERR, "\n");
    return;
}
```

7.1.4 Processor SDK 7.1 SBL 示例

zip 文件中附有一个使用上述示例代码对 Processors SDK 7.1 中的 SBL 引导流程进行修改的示例。

所附 zip 文件中的示例代码基于 Processor SDK 7.1 中提供的 Processor SDK 存储器映射。可能需要针对受测平台定制防火墙存储器范围。

若要查看更改，可以比较 zip 文件中的两个目录：

- ti-processor-sdk-rtos-j721e-evm-07_01_00_11_baseline
- ti-processor-sdk-rtos-j721e-evm-07_01_00_11_firewalls

重要声明和免责声明

TI 提供技术和可靠性数据 (包括数据表)、设计资源 (包括参考设计)、应用或其他设计建议、网络工具、安全信息和其他资源, 不保证没有瑕疵且不做任何明示或暗示的担保, 包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任: (1) 针对您的应用选择合适的 TI 产品, (2) 设计、验证并测试您的应用, (3) 确保您的应用满足相应标准以及任何其他安全、安保或其他要求。这些资源如有变更, 恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务, TI 对此概不负责。

TI 提供的产品受 TI 的销售条款 (<https://www.ti.com/legal/termsofsale.html>) 或 [ti.com](https://www.ti.com) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

邮寄地址: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2021, 德州仪器 (TI) 公司

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司