

trust zone之我见

原创

hovan-邓永坚

于 2015-01-08 19:52:53 发布


16274

收藏

52

版权

分类专栏: Trustzone

 Trustzone

专栏收录该内容

0 订阅 1 篇文章

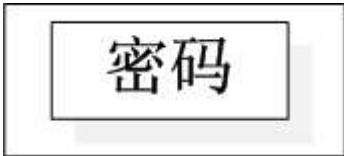
订阅专栏

老板交待任务，这个星期我都在研究trust zone的东东，之前有看过代码，但没有深入了解！

好吧，这次看来我要跟它杠上了。

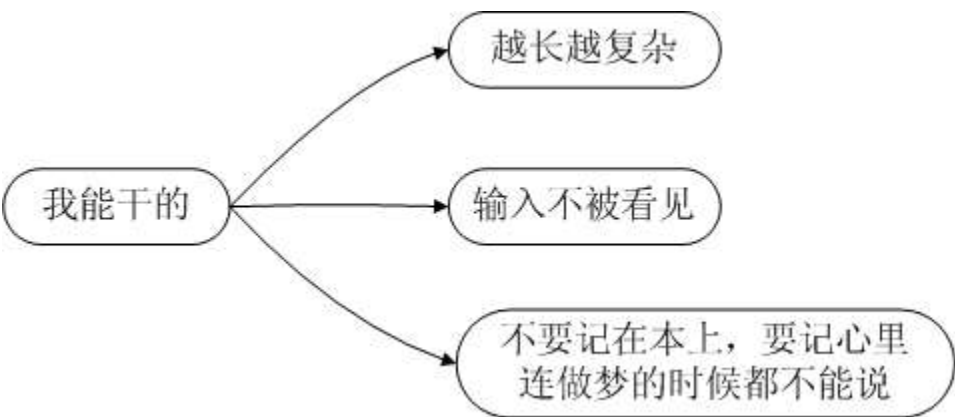
网上有很多资料，但很多讲得太抽象，至少对门外汉来说有些难以理解，我估计有些文单可能翻译过来的吧，有些拗口。

在介绍trust zone之前！我们来看两个字，慢慢引导大家trust zone与之前的安全方式有何不同？



好吧，太熟悉了，你有多少密码？QQ密码有没有？银行密码有没有？支付宝密码有没有？

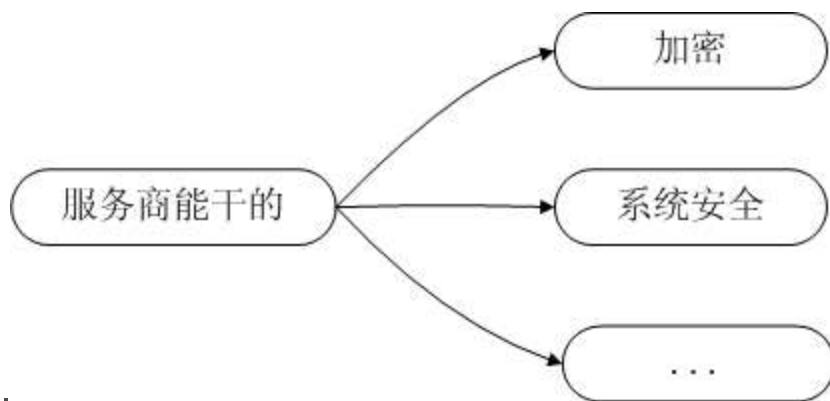
那你怎么保证你的密码安全？



that is all, 够了吗?

还记得11年的CSDN密码事件吗? 我也是受害者。

事实证明, 还要靠服务商! 他们把我们的密码记录在磁盘上, 一但被黑客读取破解。



系统这块 - - - > 比如各种防火墙, 各种安全机制。

密码加密 - - - > 比如用各种方法加密, 越复杂越好, 密钥鬼长鬼长的。

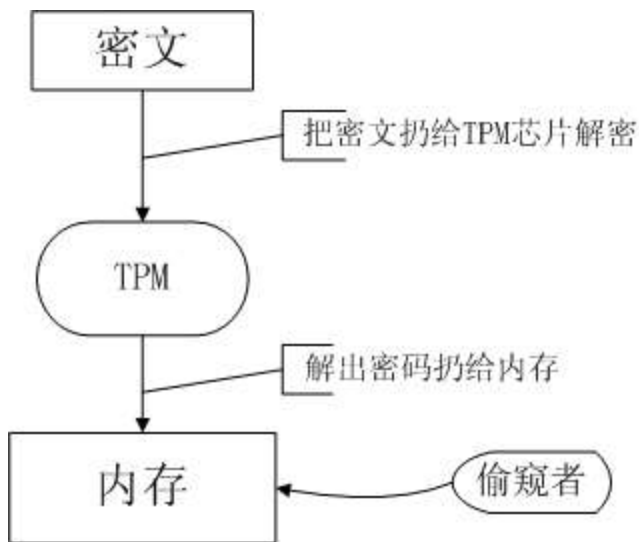
其实我们知道, 是软件就有漏洞, 迟早被破解。

所以人们转向安全芯片的研制:

即: TPM(Trusted Platform Module)

就是加密解密动作在芯片中进行, 甚至可把信息存储在芯片里。

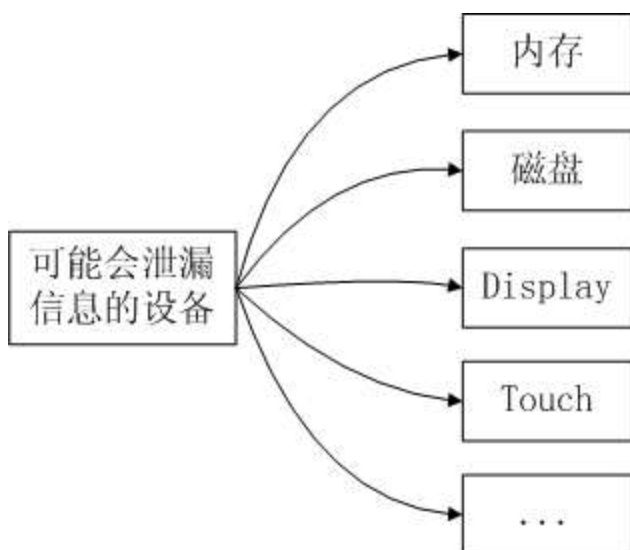
理论上来说只有芯片才能解密。但是TPM没有办法保护运行时攻击, 比如黑客在你运行进破解, 直接去内存读你解密过的东西, 这样TPM就形同虚设了。



那么下面的trust zone则完全不同，它从硬件角度做到安全。即受它保护的硬件，就算黑客root了你的设备也没办法访问的，只有生产者自己写的trust app才能访问。

而且secure boot技术保证了别人没办法篡改你的image。

从下面图可知：



入侵系统后，通常喜欢从内存，硬盘获取信息，有些木马还能通过截取你的touch或者display内容获取信息。

这样你防不胜防，除非你不要开机。

什么是trust zone?

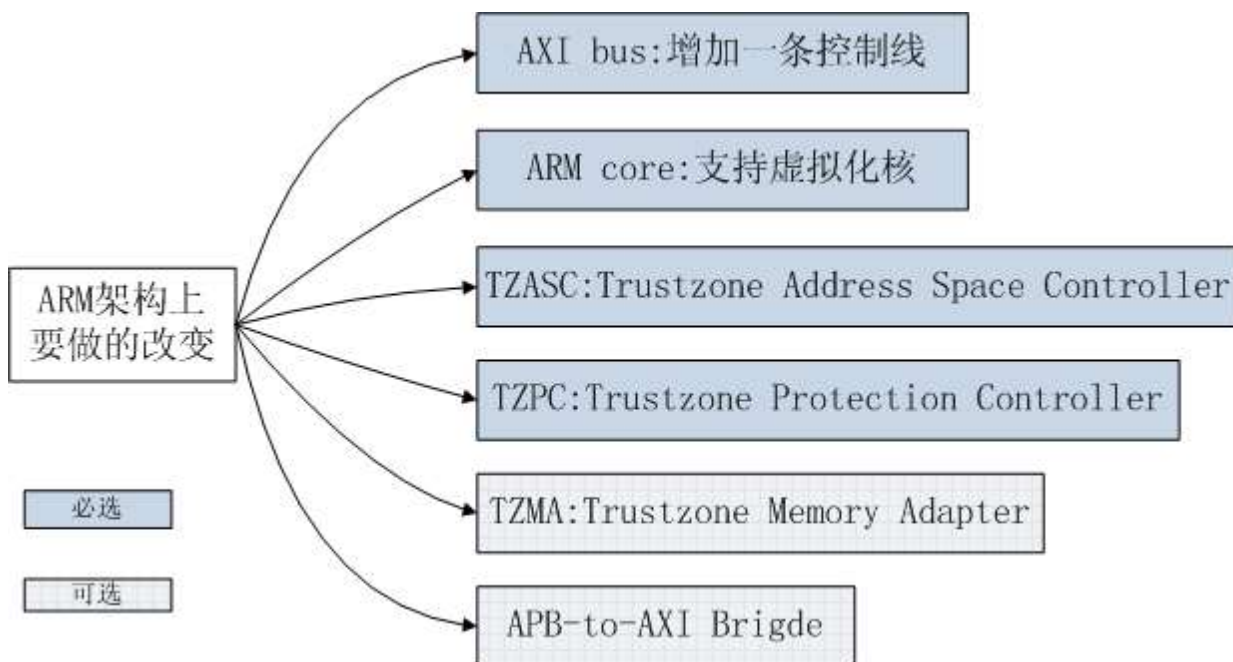
Trust zone 是ARM内核中新添加一种架构，从ARM v6KZ开始。

支持这种功能的CPU会跑在两个世界，普通世界/安全世界。

android跑起后CPU跑在普通世界，运行的是普通世界的APP，当SMC系统调用触发进入安全世界时，CPU跑在安全世界，运行安全世界的APP，安全世界APP里所用到的资源，包括内存，cache,touch,display，普通世界的app是不能够访问的，攻击者没办法拿到敏感信息

那trust zone怎么做隔离？

ARM架构上：



第一：the core AXI bus：AXI总线，增加一条控制线。

第二：ARM core,可支持虚拟化核

第三：Trust zone Address Space controller:TZASC

第四：Trust zone protection controller:TZPC

TZMA，APB-to-AXI是可选的看SOC是否支持保护外设功能

软件上：

软件上就是基于第二点可虚拟化核心，加上SMC系统调用，使CPU进入安全世界，跑安全世界的APP。

基于上述架构加上SMC调用就可以做安完全隔离了。

先谈谈AXI总线

是内存，片内静态RAM ROM， 外设隔离的基础。

主要原理是：AXI总线上每个读写信道都增加了一个额外的控制信号

AWPROT[1]：总线写事务控制信号---低电平为安全写事物， 高电平为非安全写事物

ARPROT[1]：总线读事物控制信号---低电平为安全读事物， 高电平为非安全读事物

当设备向总线提出读写事物请求时必须将控制信号发送到总线上。总线根据这个信号和CPU当前的世界来判断能否读写。防止非安全程序/设备读写安全设备。

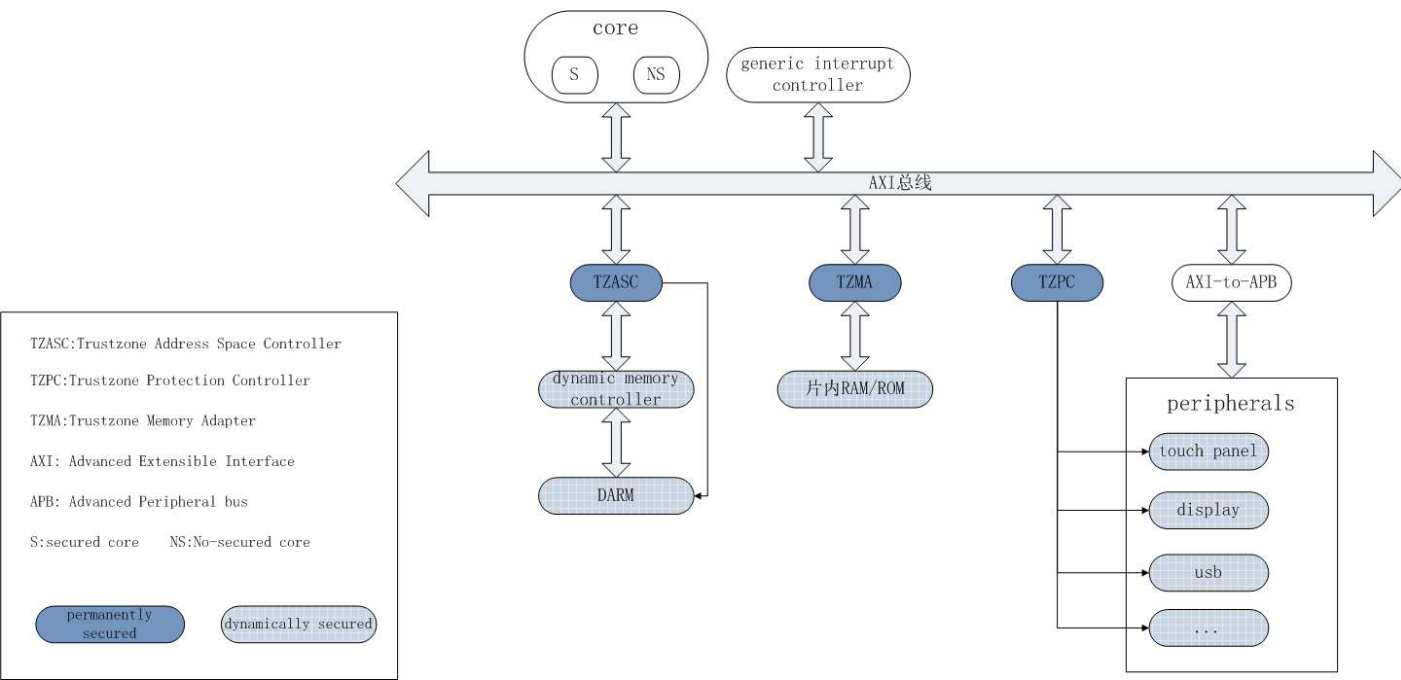
基于AXI总线， 内存， 片内静态RAM ROM是如何隔离的？

TrustZone通过两个设备来保障物理内存的安全

一个是TrustZone地址空间控制器（TZASC）

一个是TrustZone存储适配器（TZMA）

如下图：



TZASC是AXI总线的主设备，用它可以把内存地址空间划分一系列的内存空间，通过运行在安全世界的软件把部分空间配置为安全、非安全的，TZASC防止非安全事物访问安全内存空间。

使用TZASC的主要目的就是AXI的从设备分区为几个安全设备，防止非安全事物访问安全设备。ARM的DMC本身不支持创建安全，非安全区，为此需要连接到TZASC上。

注：ZASC只用来支持存储映射设备，不能用于块设备，比如NAND FLASH

TZMA是AXI总线的主设备，用它来划分片内RAM，ROM的安全区间

基于AXI总线，外设是如何隔离的？

看上图，由于APB总线没有AXI总线有trustzone安全相关的控制信号，需要APB-to-AXI桥负责，外设还是与APB连接，APB-to-AXI桥有上TZPCDECPORT信号输入，用它来决定配置外设是安全的，非安全的。APB-to-AXI桥杜绝非安全事物访问外设

TZPCDECPORT输入信号可以在SoC设计时静态地设置，也可以通过对TrustZone保护控制器（TZPC）进行编程，在程序运行时动态地设置，也就是说通过TZPC可能动态配置外设是安全的，非安全的。

另外：cache和内存为了支持trustzone安全策略，需要做些扩展。

cache的tag都增加了NS位，用于标识这一行的安全状态，NS = 0这一行处于安全状态，NS = 1这一行处于非安全状态。

MMU的TLB的tag增加NSTID位，功能与NS一样

现在已经了解trustzone保护内存外设的基本思想。

从上面已经知道，只有安全世界才能起保护作用，那如何进入安全世界的呢？

引入特殊机制 - - 监控模式，负责不同执行环境切换。

如下图：普通世界是如何进入安全世界得到服务的。

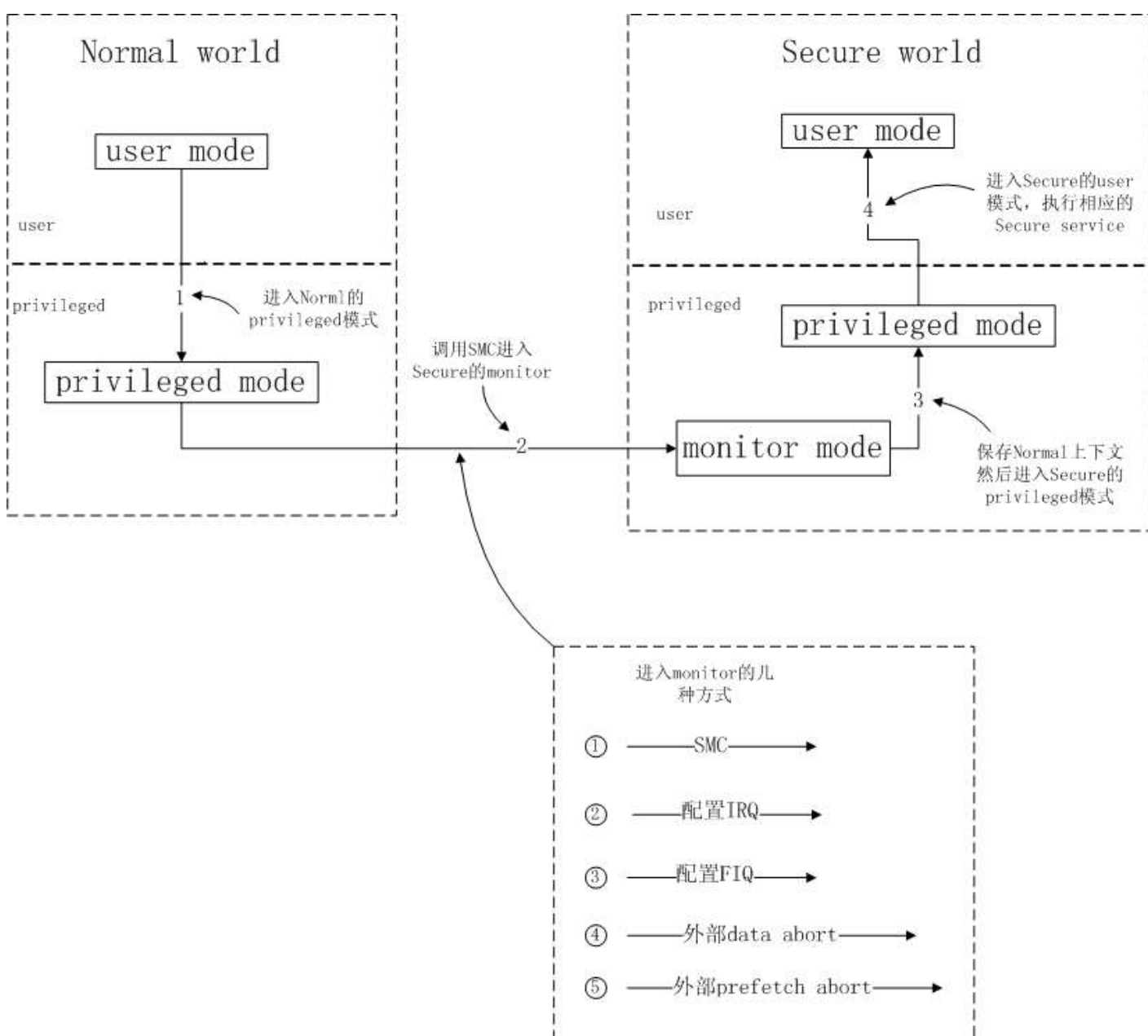
第一：运行在普通世界用户模式的APP进入特权模式

第二：该模式下调用SMC进入安全世界的monitro模式

第三：安全世界的monitor保存普通世界的上下文，然后进入安全世界的特权模式

第四：然后进入安全世界的用户模式，执行相应的安全服务

从下面小框框得知：除了软件调用SMC，还有外部各种异常都可以进入monitor模式，不过这些异常需要配置才能使用



上面通过SMC进入Secure world，那么ARM处理器如何知道当前是什么状态？

支持trustzone的ARM处理器的协处理器CP15有个安全配置寄存器（SCR），该寄存器有个NS位，这个NS位指明当前系统状态。

如果NS=0，系统处于安全状态，NS=1，系统处于非安全状态。当系统处于monitor模式，不管NS=0,1，都可以访问所有安全环境的资源，这个NS不仅影响CPU内核，内存子系统，还影响外设工作，是支持trustzone功能的关键扩展。从上图可知，系统的安全状态与系统的应用模式和特权模式无关，也就是说应用程序运行在非安全态，不管是用户模式还是特权模式，都是属于非安全世界。反之安全世界的应用程序也有应用模式与特权模式。两个世界都有应用和特权模式，每种模式所具有的权限是不同的，NS位只能被运行在安全世界处于特权模式的软件改变，系统在非安全状态时不能访问SCR寄存器。

关于进入monitor模式方式的详细说明

如下图：



- 1: SMC是一个特殊指令，类似于软件中断指令（SWI），通过它来进入mointor模式
- 2: 外部中止预取指令外部中止和数据中止，外部中止是访问存储系统时发生，但不被MMU所检测到异常，通常发生在普通世界访问安全世界资源时发生。
- 3: 中断，包括FIQ，IRQ。

其中第一种进入monitor模式是无条件的，后面两种情况依赖于SCR寄存器相关配置

* EA, =0, 表示发生外部中止时处理器进入中止模式, =1,表示发生外部中止时处理器进入monitor模式。

* IRQ, =0, 表示发生IRQ时处理器进入中止模式, =1,表示发生IRQ时处理器进入monitor模式。

* FIQ, =0, 表示发生FIQ时处理器进入中止模式, =1,表示发生FIQ时处理器进入monitor模式。

我们知道了如何通过monitor模式，从而进入安全世界，那如何从安全世界返回到普通世界呢？

答：也得从monitor模式切回来，虽然运行在安全世界的软件有更改SCR寄存器NS位的权利，但不建议这么做。因为如果安全环境的软件在非monitor模式下直接将SCR的NS位设置为1，则系统直接进入非安全状态，这使得非安全世界有看到正在流水线的指令，以及正在寄存器中的数据的可能，如果这些指令和数据都是敏感信息的话，这就给系统带来安全威胁，因为通常只有monitor可能直接修改主SCR的NS位。

好吧，现在我们已经对trustzone的工作原理有了大致了解，下面还要补充几个比较细的知识点

关于trustzone的中断控制器

在ARM传统的向量中断控制器（VIC）基础上，添加了trustzone中断控制器（TZIC）。TZIC，VIC控制器通过菊花链的方式连接组成两级中断控制系统，目的是做到普通中断与安全中断的隔离，安全中断不能被普通世界捕获。TZIC是第一级中断控制器，所有中断源的中断请求都在连在TZIC上的，它最先截获设备的中断请求，通过对TZIC的TZICIntSelect寄存器进行编程，可以对中断源产生的中断类型进行设置。如果TZICIntSelect中的某一位 = 1，则相应中断源请求被配置为FIQ中断，如果 = 0，则该中断源的中断请求将交由第二级中断控制器VIC处理。凡是由TZICIntSelect为FIQ中断的中断源提出的中断请求将绕过VIC而直接由TZIC处理，没有被TZICIntSelect配置为FIQ中断的中断源具体会被设置为FIQ还是IRQ，这将由VIC的VICIntSelect寄存器来决定，当然一般情况下应该配置为IRQ，如果被配置为FIQ，中断请求又将p被反馈给TZIC。

TZICIntSelect寄存器复位值 = 0，也就是说默认所有中断都交给VIC处理，这样对不支持trustzone的软件系统来说，可以把TZIC看作完全透明的。

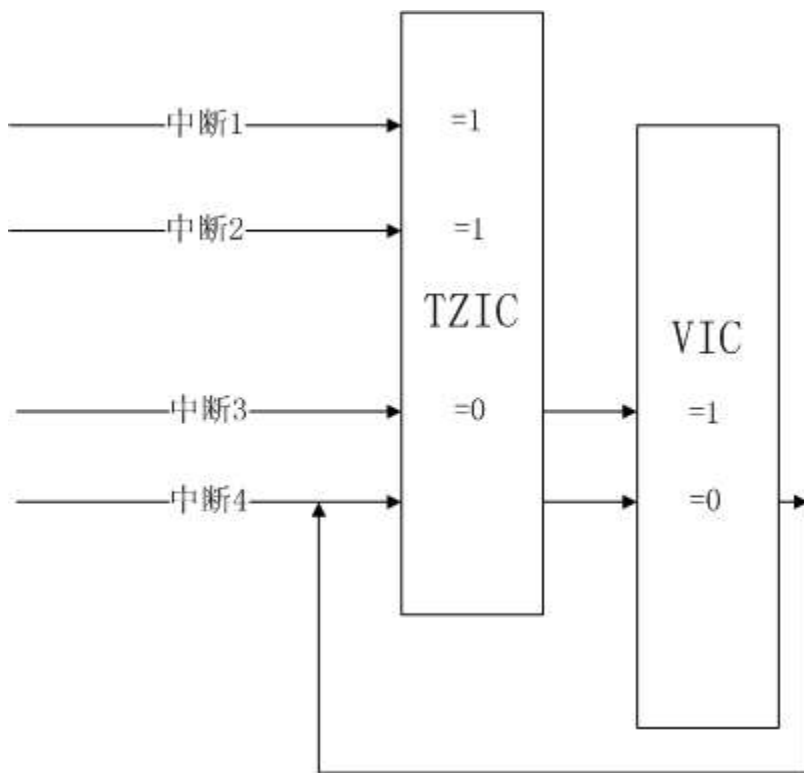
如下图所示：

中断1，中断2在TZICIntSelect都设置为1，所以直接由TZIC处理了，

中断3在TZICIntSelect设置为0，交由VIC处理

中断4在TZICIntSelect设置为0，交由VIC处理，但在VICIntSelect也设置为0，又交还给TZIC了。

最后总结：中断1，中断2，中断4属于FIQ，中断3属于IRQ。



关于trustzone的异常向量表

带trustzone的ARM处理有三个异常向量表，一个普通世界的异常向量表，一个安全世界的异常向量表，一个monitor的异常向量表。

在系统开机时，安全世界的异常向量表基地址是0x00000000或者0xffff0000，取决于处理器输入信号VINTHI，其它两个向量表的基地址开机是未定义，使用前必须软件设置。

与以前的普通ARM处理器不同，每个异常向量表的位置在运行时可以动态移动，将新的异常向量表基地址写入CP15的VBAR寄存器即可，monitor的向量表基地址由monitor的异常向量表基地址寄存器指定。

另外普通世界与安全世界的向量表基地址除了与VBAR有关，还与处理器的V位有关，v=1，则向量表基地址采用高地址，而与VBAR无关。普通世界与安全世界的V位是独立的。

好了，基本讲完，我讲不是很详细，目的是让初学者也能按照我的思路慢慢了解trustzone，更专业的知识可以参考其它资料。

谢谢