

## 快速链接:

👉 👉 👉 [ARMv8/ARMv9架构入门到精通-\[目录\]](#) 👉 👉 👉

- 付费专栏-付费课程 **【购买须知】** :
- 联系方式-加入交流群 ---**联系方式-加入交流群**
- 个人博客笔记导读目录(全部)

### 1、同步异常和异步异常的概念

具备以下3个行为的称之为同步异常:

- The exception is generated as a result of direct execution or attempted execution of an instruction.
- The return address presented to the **exception handler** is guaranteed to indicate the instruction that caused the exception.
- The exception is precise

其实就是说:

- 异常是由执行或尝试执行指令产生的
- 产生异常的那个位置是确定的, 即每次执行到“那个指令处”就会产生
- 异常是precise的

具备以下3个行为的称之为异步异常: :

- The exception is not generated as a result of direct execution or attempted execution of the instruction stream.
- The return address presented to the exception handler is not guaranteed to indicate the instruction that caused the exception.
- The exception is imprecise.

其实就是说:

- 异常**不是**由执行或尝试执行指令产生的
- 产生异常的那个位置**不是**确定的, 即不知道执行到哪里, 就产生了异常
- 异常是imprecise的

## 那么precise 和 imprecise 又是什么意思呢？

An exception is described as **precise** when the exception handler receives the PE state and memory system state that is consistent with the PE having executed all of the instructions up to but not including the point in the instruction stream where the exception was taken, and none afterwards.

An exception is described as *imprecise* if it is not precise.

比较绕、比较难懂，咱们换一个说法：按照预期产生的异常称之precise，反之imprecise

## 2、系统中有哪些同步异常？

- 尝试执行UNDEFINED指令产生的任何异常，包括：
  - (1)、尝试在不适当的异常级别执行指令。
  - (2)、当指令被禁用时尝试执行指令。
  - (3)、尝试执行尚未分配的指令位模式。
- 非法执行状态异常。这些是由尝试执行指令引起的 `PSTATE .IL` 为 1，(详细可参考D1-2486 页上的AArch64 状态的非法返回事件)
- 使用未对齐的 SP 导致的异常。
- 尝试使用未对齐的 PC 执行指令导致的异常。
- 由异常生成指令SVC、HVC或SMC引起的异常。
- 尝试执行系统寄存器定义为被捕获到更高的异常级别。(详细可参考可配置的指令使能和禁止，在D1-2510 页)
- 由内存地址转换系统生成的指令中止与尝试相关联从产生故障的内存区域执行指令。
- 内存地址转换系统生成的数据中止与尝试读取或写入产生故障的内存。
- 由地址未对齐引起的数据中止。
- 如果实施FEAT\_MTE2，则由标记检查故障引起的数据中止。。
- 所有调试异常：
  - (1)、Breakpoint Instruction exceptions.
  - (2)、Breakpoint exceptions.
  - (3)、Watchpoint exceptions.
  - (4)、Vector Catch exceptions.
  - (5)、Software Step exceptions.
- 在支持捕获浮点异常的实现中，由捕获的IEEE 浮点异常引起的异常
- 在某些实现中，外部中止。外部中止是失败的内存访问，包括访问地址转换期间发生的内存系统的那些部分。

## 3、Serror的理解

Serror（也称System Error），是异步异常的一种，一般是来自 `External aborts`，当memory system访问时bus上产生的 `External aborts`。例如

- 访问内存被TZC挡住时，bus会返回的一个异常

There are three types of physical interrupt:

- SError (also described as a System Error).
- IRQ.
- FIQ.

There are three types of virtual interrupt:

- vSError (also described as a Virtual System Error).
- vIRQ.
- vFIQ.

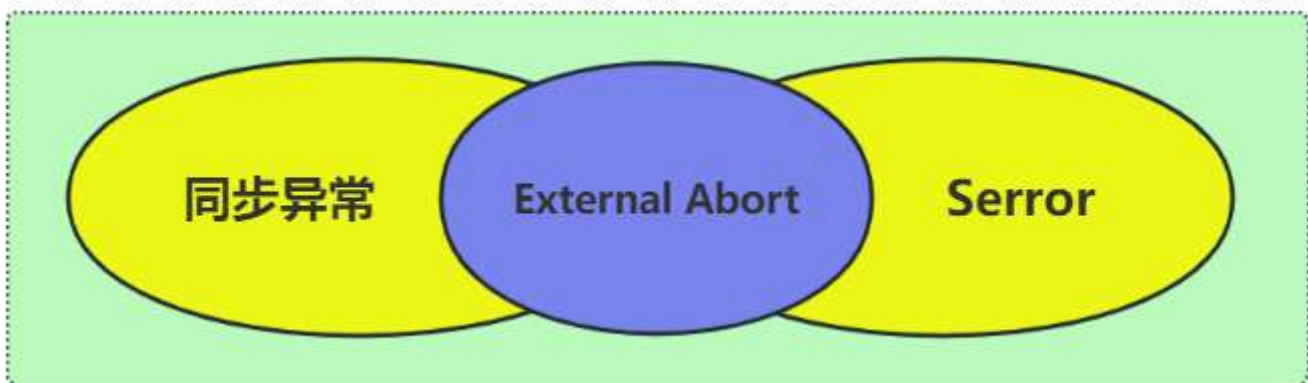
The physical SError interrupt is often used, amongst other things, for communicating External aborts from the memory system that are to be taken asynchronously.

For an External abort generated by the memory system that is taken asynchronously using the SError interrupt, the SError interrupt always behaves as an edge-triggered interrupt. For any other sources of SError interrupts, it is IMPLEMENTATION DEFINED whether they are edge-triggered or level-sensitive. CSDN @代码改变世界ctw

注意，像MMU产生的异常，它属于 **internal abort**，它是同步异常。像未定义指令异常，它也是同步异常

#### 4、External abort的理解

External Abort，可以同步异常，也可以是SError



例如：

- Instruction Abort 可能是内部同步异常，也可能是同步External Abort
- Data Abort 可能是内部同步异常，也可能是同步External Abort  
(即 External abort 可以属于同步异常哦)

0b100000	Instruction Abort from a lower Exception level. Used for MMU faults generated by instruction accesses <b>and</b> synchronous <b>External</b> aborts, including synchronous parity or ECC errors. Not used for debug-related exceptions.
0b100001	Instruction Abort taken without a change in Exception level. Used for MMU faults generated by instruction accesses <b>and</b> synchronous <b>External</b> aborts, including synchronous parity or ECC errors. Not used for debug-related exceptions.

0b100100	Data Abort exception from a lower Exception level. Used for MMU faults generated by data accesses, alignment faults other than those caused by Stack Pointer misalignment, <b>and</b> synchronous <b>External</b> aborts, including synchronous parity or ECC errors. Not used for debug-related exceptions.
0b100101	Data Abort exception taken without a change in Exception level. Used for MMU faults generated by data accesses, alignment faults other than those caused by Stack Pointer misalignment, <b>and</b> synchronous <b>External</b> aborts, including synchronous parity or ECC errors. Not used for debug-related exceptions.

#### 4、instruction abort、data abort

在aarch64架构中，instruction abort、data abort已然变成了同步异常中的一种。不再是单独的异常模型了。所以当出现instruction abort、data abort时，那么一定是同步异常，可能是internal abort，也可能是external abort，但终归是同步异常。

## 社群



ARM-TEE-ATF-SOC群(一)



ARM-TEE-ATF-SOC群(六)



ARM-TEE-ATF-SOC群(七)



ARM-TEE-ATF-SOC群(三)



ARM-TEE-ATF-SOC群(五)



ARM-TEE-ATF-SOC群(四)



ARM-TEE-ATF-SOC群(二)



T03-Arm二期课程交流群



T02-Arm一期课程交流群



T04-Trustzone/TEE课程交流群



T06-Secureboot课程交流群1299



T01-Arm一期课程交流群CSDN



T05-Secureboot课程交流群499



PAC&amp;BTI&amp;MTE三剑客课程群

(想进社区的加v: arm2023, 备注: CSDN进群)

Armv8/Armv9架构从入门到精通, Armv8/Armv9架构从入门到精通 (一期), Armv8/Armv9架构从入门到精通 (二期) Armv8/Armv9架构从入门到精通 (三期), Arm一期、Arm二期、学习资料、免费、下载, 全套资料, Secureboot从入门到精通, secureboot训练营, ATF架构从入门到精通、optee系统精讲、secureboot精讲, Trustzone/TEE/安全快速入门班, Trustzone/TEE/安全标准版, Trustzone/TEE/安全高配版。全套资料。周贺贺, baron, 代码改变世界, coding\_the\_world, Arm精选, arm\_2023, 安全启动, 加密启动 optee、ATF、TF-A、Trustzone、optee3.14、MMU、VMSA、cache、TLB、arm、armv8、armv9、TEE、安全、内存管理、页表, Non-cacheable,Cacheable, non-shareable,inner-shareable,outer-shareable, optee、ATF、TF-A、Trustzone、optee3.14、MMU、VMSA、cache、TLB、arm、armv8、armv9、TEE、安全、内存管理、页表...