关于本手册

Processsing element 简称 PE

AArch32，包括了A32和T32

AArch64，只有A64指令集

PE操作的状态机，包括当前的异常级别和安全级别

异常模型

进程间的模型，支持AArch64和AArch32的转换

内存模型

编程者的模型，主要是寄存器什么的

Advanced SIMD指令以及浮点指令

单精度半精度双精度浮点操作

各种浮点数之间的转换

整数浮点的向量操作

安全模型

虚拟化模型

Debug架构

* 1. 各个部分的内容

**Part A**概论

**Part B** 描述 AArch64 执行状态的应用进程级别视图，即来自 EL0 的视图。它描述了编程者模型和内存模型的应用进程级别视图。

**Part C** A64指令集，包括每条指令在EL0下的效果，也就是说非特权级下的执行，也包括它的使用限制，以及在高特权级下的效果。这部分对于编译器汇编器以及其他的生成ARM机器码的人来说是非常重要的

**Part D** AArch64执行状态的系统级视图的描述。包括系统寄存器的细节，这些在EL0状态无法访问，以及在系统级别的视图下的编程者模型和内存模型。以及包括了self-hosted debug。

**Part E** EL0下的AArch32执行状态的应用视图，也就是在User mode。

**Part F**描述了T32和A32，其他和Part C类似

**Part G** AArch32下和Part D类似的内容

**Part H** Debug描述，包括external Debug，以及configuration，breakpoint，以及watchpoint的支持，以及a Debug Communications Channel (DCC) to a debug host

后面的Part看描述貌似不重要。

本次阅读的基础目标是完成A-D，只考虑AArch64。扩展阅读目标是剩余部分，包括了A32和T32的内容。

唔，8k行内容即便只读partA-D，仍然很麻烦，要好几千页。

Part A

这个手册并不试图规定如何实现，也不限制在这些规定之外的一些实现

而且ARM体系结构也包括debug，self-hosted debug和external debug

包括Trace Unit。The Embedded Trace Macrocell Architecture Specification. Self-hosted trace。

RISC架构

具有大的统一寄存器文件。

Load/store架构

简单的寻址模式。

同样定义了PE和内存的交互，包括cache。内存转移系统