

中断 APIC 读书笔记

Hollow Man

在计算方面，英特尔的高级可编程中断控制器（APIC）是中断控制器系列。顾名思义，APIC 比英特尔的 8259 可编程中断控制器（PIC）更先进，尤其能够构建多处理器系统。它是旨在解决多处理器计算机系统中中断路由效率问题的几个架构设计之一。

APIC 是一种拆分架构设计，其本地组件（LAPIC）通常集成到处理器本身中，并在系统总线上集成可选的 I/O APIC。第一个 APIC 是 82489DX，它是一个离散芯片，兼具有本地和 I/O APIC 的功能。82489DX 实现了与英特尔 486 和早期奔腾处理器对称多处理器（SMP）系统的构建；例如，参考双向 486 SMP 系统使用了三个 82489DX 芯片，其中两个用作本地 APIC，一个用作 I/O APIC。从 P54C 处理器开始，本地 APIC 功能集成到英特尔处理器的硅中。第一个专用的 I/O APIC 是英特尔 82093AA，该系统适用于基于 PII X3 的系统。

标准 ISA 兼容的中断控制器（位于 PII X3 中）旨在用于单在处理器系统中，I/O 高级可编程中断控制器（IOAPIC）可以在单处理器或多处理器系统。IOAPIC 提供多处理器中断管理和合并了所有处理器之间的静态和动态对称中断分配。

在具有多个 I/O 子系统，每个子系统可以有自己的一组中断。在单处理器系统中，IOAPIC 的专用中断总线可以减少标准上的中断等待时间中断控制器（即，与中断确认周期的传播相关的延迟使用标准中断控制器方法的多条总线）。中断可由标准控制 ISA 在 PII X3，IOAPIC 单元或混合模式下均兼容 ISA，这两种标准 ISA 都兼容使用兼容的中断控制器和 IOAPIC。选择哪个控制器响应中断由中断控制器的编程方式决定。请注意，程序员有责任确保两个中断控制器均未处理相同的中断输入信号。

在系统级别，APIC 由两部分组成（图 2.0），其中一个位于 I/O 子系统中（称为 I/O 子系统）。IOAPIC），另一个在 CPU 中（称为 Local APIC）。本地 APIC 和 IOAPIC 通过专用 APIC 总线。IOAPIC 总线接口由两个双向数据信号（APICD [1: 0]）和一个时钟输入（APICCLK）。

CPU 的本地 APIC 单元包含必要的信息，以确定其处理器是否应接受在 APIC 总线上广播的中断。本地单元还提供本地挂起的中断，嵌套和屏蔽中断，并处理与其本地处理器（例如，INTR / INTA / EOI）的所有交互协议）。本地单元还向其本地处理器提供处理器间中断和定时器。这每个处理器的处理器与其本地 APIC 的寄存器级接口都是相同的。

IOAPIC 单元由一组中断输入信号，一个 24 条目乘 64 位中断重定向表，可编程寄存器，以及用于通过 APIC 总线发送和接收 APIC 消息的消息单元。输入/输出设备通过置位 IOAPIC 的中断线之一将中断注入系统。IOAPIC 在重定向表中选择相应的条目，并使用该条目中的信息来格式化中断请求消息。重定向表中的每个条目都可以单独编程以指示边沿/电平敏感中断信号，中断向量和优先级，目标处理器以及如何选择处理器（静态或动态）。表中的信息用于将消息发送到其他 APIC 单元（通过 APIC 总线）。

IOAPIC 包含一组可编程寄存器。两个寄存器（I/O 寄存器选择和 I/O 窗口寄存器）位于 CPU 的存储空间中，用于间接访问其他 APIC 寄存器，例如：在第 3.0 节，寄存器描述中进行了介绍。版本寄存器提供了 IOAPIC。使用 IOAPIC 的物理名称的 ID 值对 IOAPIC ID 寄存器进行编程。写入 IOAPIC ID 寄存器后，此 ID 会加载到 ARB ID 寄存器中，并在总线期间使用仲裁。

中断号或向量并不表示要发送的特定优先级。IOAPIC 不断轮流轮询 24 个中断，一次轮询一次。首先轮询的待处理中断是发送的中断。

但是 APIC 系统的实现中存在许多已知错误，特别是 8254 的连接方式。有缺陷的 BIOS

可能无法正确设置中断路由，或提供不正确的 ACPI 表和英特尔多处理器规范（MPS）表。

当操作系统不能正确支持时，APIC 也可能是系统故障的原因。在较旧的操作系统中，I/O 和本地 APIC 通常必须禁用。虽然由于对称多处理器和多核系统的流行，这已经不可能了，但固件和操作系统中的错误现在很少发生。

与前两个相比，MSI 显著减少了中断等待时间几代英特尔中断体系结构。好处超出了通过消除中断延迟来降低 CPU 利用率。通过使用消息信号中断来减少中断等待时间 32107021 岁 CPU 确定需要处理哪些中断所花费的时间（由轮询设备和屏蔽中断控制器）。嵌入式开发人员考虑将英特尔®架构作为解决方案或正在开发中应该完全采用 MSI 模型进行中断传递和服务确保不仅为其解决方案提供最佳的 IO 性能，而且确保最大程度的用户应用程序和其他中断的 CPU 余量。

总而言之，MSI 与以前的版本相比，它为嵌入式开发人员提供了以下关键优势中断架构：

- 增加中断数量以支持更多设备和外设。
- 大大减少了从需要维修设备到维修的延迟当 CPU 开始维修设备时。
- 简化的电路板设计：无需中断控制器（IO-APIC / PIC）。
- 灵活的中断优先级分配方案。
- 跨 CPU 的中断负载平衡。设备可以将中断定向到特定核心以利用通用缓存并确保相等所有 CPU 上的工作负载。