

Keystone 系列 DSP 中断子系统结构和中断映射原理

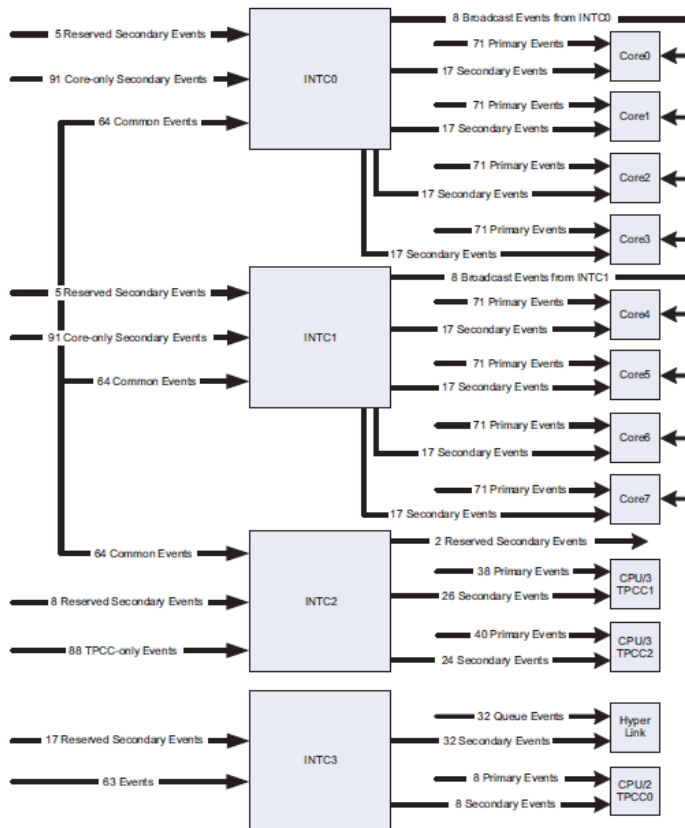
King Wang

本文以 Keystone 系列的 C6678 Soc 为例，解释 Keystone 系列 DSP 的中断子系统架构，详细说明中断映射的配置方法和原理。

1. 中断子系统架构概述

TI 的 Keystone 系列 DSP 的中断子由两部分组成，芯片中断控制器（INTC）和 DSP 核（以下简称 CorePac）中断处理控制器（CorePac Interrupt Control）。INTC 共有四个，INTC0 主要负责 Core0 到 Core3，INTC1 主要负责控制 C4 到 C7，INTC2 负责控制 EDMA3 的 TPCC1 和 TPCC2，INTC3 负责 EDMA3 的 TPCC0 以及 Hyperlink 的中断。CorePac 内部的中断控制器位于 CorePac 内部，主要负责将外部事件转换为 CorePac 内部的中断信号，在 DSP 的 Core0 到 Core7 各有自己的核内中断控制器。

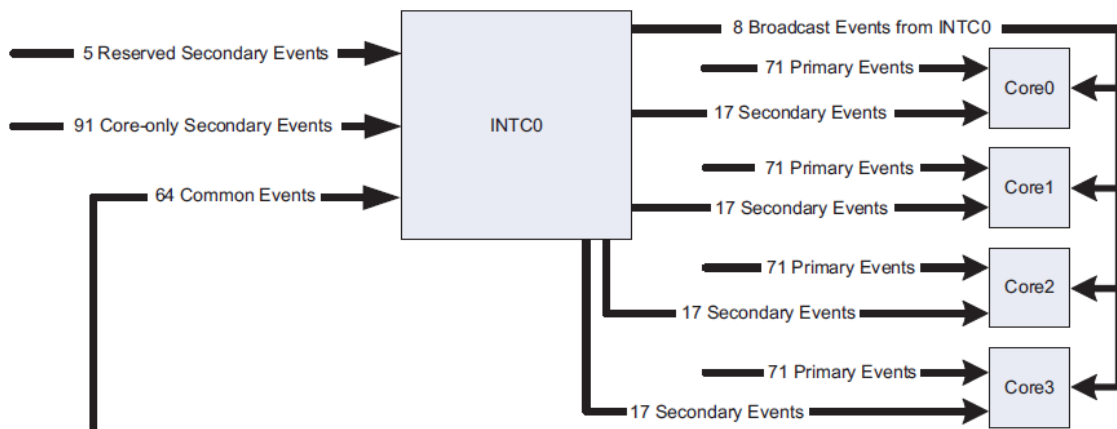
下图是整个 Soc 片内中断处理子系统框图。



2. 中断控制器（INTC）

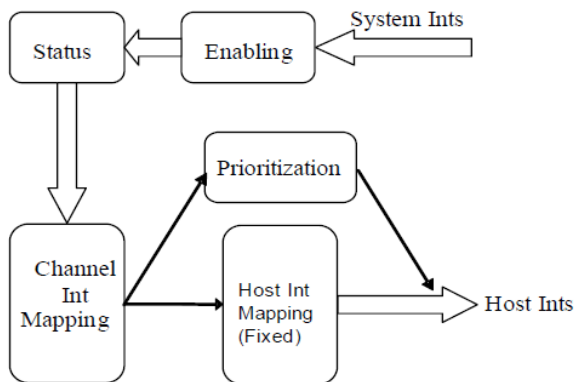
INTC 的主要功能为将系统事件（system interrupt）映射为 CorePac 内部中断控制器可以处理的主机事件（host interrupt）。由于 CorePac 中断控制器最多只能处理 128 个输入事件，那么 INTC 的一个重要的功能就是将多个系统事件映射为多个或者单个主机事件。

我们将 INTC0 的输入和输出单独拿出来分析。INTC0 的左边输入共有 $5+91+64=160$ 个系统事件，而它的输出给 Core0 的为 $17+8=25$ 个主机事件（手册上描述的是二级事件，但转换后的这 17 个二级事件与 CorePac 里面可以处理的事件是一一对应的）。



从上图来分析，INTC0 的主要作用是将左边的 160 个事件，转换为右边的 17+8 个事件。

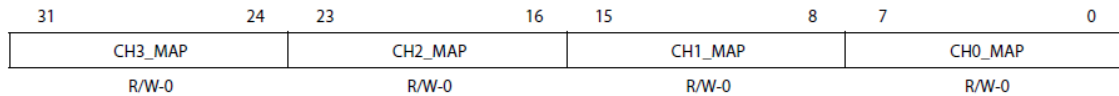
INTC 的事件映射功能，主要是将系统事件映射到 channel 中，然后每个 channel 又映射到主机事件中，因为 CorePac 能直接处理主机事件。因此，整个映射分为两部分，系统事件映射和信道映射。如下图所示。



系统事件映射

信道中断映射寄存器（CH_MAP_REGx）寄存器保存了每个系统事件对应的信道。如下图，1 个 32bit 的 CH_MAP_REG 寄存器可以配置 4 个系统事件。根据寄存器的地址映射关系可以得出，总共有 256 个 CH_MAP_REG 寄存器，因此最多系统可以支持 $4 \times 256 = 1024$ 个系统事件。由于每个系统事件只能配置 8 个 bit 信息，因此最多可以映射的信道个数为 256 个。

Figure 2-15 Channel Interrupt Map Register (CH_MAP_REGx)



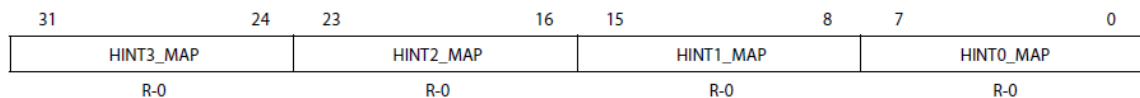
Legend: R = Read only; -n = value after reset

信道映射

也存在一个主机中断映射寄存器（HINT_MAP_REGx），这个寄存器完成 INTC 的信道到 CorePac 中断控制器的主机事件的映射。如下图所示，每个 32bit 的寄存器保存了 4 个信道的映射值，总共有 64 个寄存器，因此可供映射的信道总数总共有 256 个。但是，在 C6678 的手册中，每个 INTC0 对应的信道只有 68 个（其余都预留），即 Core0 到 Core3 每个对应 17 个信号。

需要说明的是，这个寄存器是不可配置的，是只读寄存器。正如前面提到了，信道和主机事件是一一对应的，不可更改。

Figure 2-16 Host Interrupt Map Register (HINT_MAP_REGx)

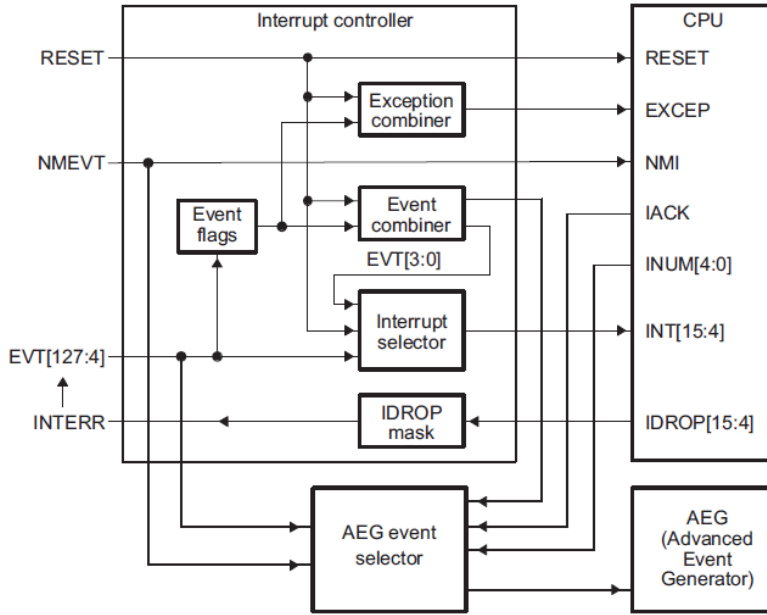


Legend: R = Read only; -n = value after reset

映射配置完成后，1024 个系统事件就可以有选择映射到 256 个 CorePae 可以处理的主机事件。

3. CorePac 中断控制器

CorePac 中断控制器是一个比较复杂的功能单位，但是如果仅从中断事件的角度来说，还是比较简单的。



如上图所示，CorePac 的中断控制器有 3 个输入来源，RESET，NMEVT 和 EVT。分别是复位信号，不可屏蔽中断信号和普通事件信号。其中普通事件总共有 128 个。在 CPU 内，可供处理的信号为 INT[15:4]，即只有 12 个可供 CPU 处理的中断信号。因此，CorePac 的主要工作就是将 128 个 EVT 转换为 12 个 INT。

CorePac 的中断控制器中的 128 个事件的组成为 3 部分，第一部分是组合事件，即 EVT0-EVT3，这部每个事件可由剩余的 124 个事件组合而成，相当与一个逻辑的或门。第二部分 INTC 信道映射过来的主机中断；剩下的为第三部分。

中断映射

这里的中断映射指的是从 128 个中断事件中选择 12 个映射到 CPU 侧的 INT 中。中断映射需要用到 CorePac 内部的中断复用寄存器（Interrupt Mux Register），如下图所示。

Figure 9-35 Interrupt Mux Register 1 (INTMUX1)

31	30		24	23	22		16
Reserved	INTSEL7				Reserved	INTSEL6	
R-0	R/W-7h				R-0	R/W-6h	
15	14		8	7	6		0
Reserved	INTSEL5				Reserved	INTSEL4	
R-0	R/W-5h				R-0	R/W-4h	

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

每个 CorePac 内部有 3 个 Mux 寄存器，每个 Mux 可以配置 4 个 INT，因此总共配置 12 个 INT。

4. 综述

综上所述，每个 CPU 可以配置的中断只有 12 个，但是最多可对应 1024 个事件。通常，一个事件的产生到 CPU 中断最长需要经过以下过程。

二级事件的中断产生过程：

- 1) 事件源产生事件
- 2) 通过 INTC 将事件转换为信道
- 3) 信道转换为主机中断
- 4) 主机中断转换为 CPU 中断

一级事件的中断产生过程

- 1) 事件源产生事件
- 2) CorePac 中断控制器将事件转换为 CPU 中断

需要说明的是，在某些加速器内部也有其自己的中断控制器，比如 NetCP 内部有中断分发器（INTD），其功能也类似 INTC，但是其位置处于 INTC 的下级。举例来说，NetCP 里面的 Gbe 如果产生一个统计器中断，需要经历以下过程：

- 1) NetCP 产生统计器中断事件
- 2) 统计器中断事件通过 INTD 映射为二级事件 MISC_INTR
- 3) 二级事件 MISC_INTR 通过 INTC 映射到信道
- 4) 映射到主机中断；
- 5) 主机中断映射到 CPU 的 INT

参考文献

所有文档都可以在 Soc 的网页下找到，比如 C6678 中，网址和文档分别为：

网址：<http://www.ti.com/product/tms320c6678>

中断子系统：TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor (Rev. D)

中断控制器 INTC：Chip Interrupt Controller (CIC) for KeyStone Devices User's Guide (Rev. A)

CorePac 中断控制器：C66x CorePac User's Guide (Rev. C)