

异常处理流的实现

CP0 的寄存器

CP0 协处理器是 P7 新引入的功能模块，我们需要用这个模块完成两个主要功能，一个是对异常进行配置，一个是记录异常的信息。CP0 有很多个寄存器用来配置或者记录，我们只需要实现其中的几个，如下所示：

寄存器	编号	功能
SR	12	配置异常的功能。
Cause	13	记录异常发生的原因和情况。
EPC	14	记录异常处理结束后需要返回的 PC。

每个寄存器都是 32 位的，我们只需要其中的几位，列表如下：

寄存器	功能域	位域	解释
SR (State Register)	IM (Interrupt Mask)	15:10	分别对应六个外部中断，相应位置 1 表示允许中断，置 0 表示禁止中断。这是一个被动的功能，只能通过 <code>mtc0</code> 这个指令修改，通过修改这个功能域，我们可以屏蔽一些中断。
SR (State Register)	EXL (Exception Level)	1	任何异常发生时置位，这会强制进入核心态（也就是进入异常处理程序）并禁止中断。
SR (State Register)	IE (Interrupt Enable)	0	全局中断使能，该位置 1 表示允许中断，置 0 表示禁止中断。
Cause	BD (Branch Delay)	31	当该位置 1 的时候，EPC 指向当前指令的前一条指令（一定为跳转），否则指向当前指令。

寄存器	功能域	位域	解释
Cause	IP (Interrupt Pending)	15:10	为 6 位待决的中断位，分别对应 6 个外部中断，相应位置 1 表示有中断，置 0 表示无中断，将会每个周期被修改一次，修改的内容来自计时器和外部中断。
Cause	ExcCode	6:2	异常编码，记录当前发生的是什么异常。
EPC	-	-	记录异常处理结束后需要返回的 PC。

当发生异常的时候，CPU 会自动将异常信息写入 CP0 的相应寄存器（如 Cause 和 EPC）。异常处理程序会访问相应寄存器，来了解异常的信息以进行异常处理。

同学们可以按规范自行设计 CP0，一个参考的 CP0 的端口声明如下：

端口	方向	位数	解释
clk	IN	1	时钟信号。
reset	IN	1	复位信号。
en	IN	1	写使能信号。
CP0Add	IN	5	寄存器地址。
CP0In	IN	32	CP0 写入数据。
CP0Out	OUT	32	CP0 读出数据。
VPC	IN	32	受害 PC。
BDIn	IN	1	是否是延迟槽指令。
ExcCodeIn	IN	5	记录异常类型。
HWInt	IN	6	输入中断信号。
EXLClr	IN	1	用来复位 EXL。

端口	方向	位数	解释
EPCOut	OUT	32	EPC 的值。
Req	OUT	1	进入处理程序请求。

异常码

在异常处理程序中，我们需要通过访问 `Cause` 寄存器的 `ExcCode` 域来获得异常的原因，在 P7 中我们需要实现的异常有这样几种（除此之外，比较常见的还有陷入，断点调试等）：

ExcCode 的编码必须遵守规范，不然在评测的时候可能会出现问題。

异常与中断码	助记符与名称	指令与指令类型	描述
0	<code>Int</code> (外部中断)	所有指令	中断请求，来源于计时器与外部中断。
4	<code>AdEL</code> (取指异常)	所有指令	PC 地址未字对齐。 PC 地址超过 <code>0x3000 ~ 0x6ffc</code> 。
		<code>lw</code>	取数地址未与 4 字节对齐。
		<code>lh</code>	取数地址未与 2 字节对齐。
		<code>lh, lb</code>	取 Timer 寄存器的值。
		load 型指令	计算地址时加法溢出。
		load 型指令	取数地址超出 DM、Timer0、Timer1、中断发生器的范围。
5	<code>AdES</code> (存数异常)	<code>sw</code>	存数地址未 4 字节对齐。
		<code>sh</code>	存数地址未 2 字节对齐。
		<code>sh, sb</code>	存 Timer 寄存器的值。

		store 型指令	计算地址加法溢出。
		store 型指令	向计时器的 Count 寄存器存值。
		store 型指令	存数地址超出 DM、Timer0、Timer1、中断发生器的范围。
8	syscall (系统调用)	syscall	系统调用。
10	RI (未知指令)	-	未知的指令码。
12	0v (溢出异常)	add , addi , sub	算术溢出。

参考资料

CP0 设计及其相关指令的实现，以及硬软件在中断处理上的协同是 P7 中最有挑战性的部分。仅阅读教程中的简要介绍远远不够，因此课程组放出一些推荐阅读的资料，希望同学能加以研究，尝试去理解其中的思路。

推荐资料列表：

1. [L13-MIPS 系统结构-V1.pdf](#)
2. [《See MIPS Run Linux》中相关章节](#)
3. [《计算机组成与设计：硬件/软件接口》中相关章节](#)
4. Google / Bing 等搜索引擎
5. 讨论区 P7 答疑帖