CP0 协处理器

CP0 模块在 P7 实验中的主要作用是实现对异常、中断的控制，具体包括 SR、Cause、EPC、PRId 四个寄存器及与中断、异常相关的处理逻辑。

与中断异常相关的四个寄存器

SR 寄存器

- IM[7:2]，即 SR[15:10]，为 6 位中断屏蔽位，分别对应 6 个外部中断。

  相应位置 1 表示允许中断，置 0 表示禁止中断。

- IE，即 SR[0]，为全局中断使能。

  该位置 1 表示允许中断，置 0 表示禁止中断。

- EXL，即 SR[1]，为异常级。

  该位置 1 表示已进入异常，不再允许中断，置 0 表示允许中断。

Cause 寄存器

- IP[7:2]，即 Cause[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。

- ExcCode[6:2]，即 Cause[6:2]，异常编码，记录当前发生的是什么异常。

EPC 寄存器

EPC 寄存器负责保存中断/异常时的 PC 值。

PRId 寄存器

通常存入处理器 ID，可以用于实现个性的编码 :)

同学们也可以阅读参考资料获得对这四个寄存器更详细的描述。

CP0 在流水线中的作用

CP0 在流水线中负责响应中断与读写相关寄存器。

CPU 在执行指令时，指令可能会在流水级的不同部位产生异常，如在 E 级可能发生算术指令异常、在 D 级可能发生跳转指令地址异常等。CPU 并不会直接处理这些异常，而是得到一个异常码 `ExcCode`（异常码表示异常的种类，编码的具体规则将在 P7 提交要求章节讲解（暂未开放）），并将 `ExcCode` 继续向后流水直至将异常码提交给 CP0。对中断的处理同理，CPU 会将得到的外部中断信号向后流水，直至提交到 CP0。也就是说，CP0 模块负责接收 CPU 产生和接收到的各种中断请求。

CP0 在接收到中断请求后，首先会对请求的合法性进行检验。对于异常产生的中断请求，CP0 在确定当前未响应中断后，响应中断请求；对于接收到的外设中断请求，CP0 在确定 SR 寄存器相应位无中断屏蔽、当前无中断且全局中断使能允许中断后，响应该中断请求。其他情况下，CP0 不会响应中断请求。

CP0 确定响应中断请求意味着 CPU 将跳转至异常处理地址 `0x00004180`，CP0 会向 NPC 模块传递一个跳转信号。在 CPU 进入异常处理程序前，CP0 需要记录当前已进入中断（将 SR 寄存器的 EXL 置位），记录当前的异常码并将指令当前 PC 存储在 EPC 寄存器中（注意，若当前执行的指令为延迟槽指令，则需要存储 PC - 4）。CPU 执行 `eret` 指令从异常返回时，CP0 模块需负责记录当前已结束中断（将 SR 寄存器的 EXL 置 0）。

对于异常产生的中断请求，该存入 EPC 寄存器中的 PC 值是清晰的；但对于外部中断，我们该向 EPC 寄存器中存入哪个 PC 值呢？为此，我们统一引入**宏观 PC** 的概念。

> 概念解释 **宏观 PC**
>
> 宏观 PC 表示整个 CPU “宏观”运行指令所对应的PC地址。所谓“宏观”指令，表示该指令之前的所有指令序列对 CPU 的更新已完成，该指令及其之后的指令序列对 CPU 的更新未完成。(宏观 PC 是否字对齐不影响评测)。 更通俗地讲，对于宏观 PC 的意义实际上是指以某一个流水级 PC 作为界限，作为输出端口输出出来，这个流水级一般是你中断信号传入(从 bridge 接入)的流水级，例如：
>
> - 如果你的中断信号在 M 级传入，并且你的中断在 M 级响应，则宏观 PC 为 M 级的 PC。
> - 如果你的中断信号在 F 级传入，并且随着流水级走到 M 级时才响应，则宏观 PC 为 F 级的 PC。

除了响应中断，CP0 模块承担相关寄存器的读写工作，我们需要在此实现 `mtc0` 和 `mfc0` 两条指令对相关寄存器的读写。

模块接口

我们给出 CP0 设计的参考接口，同学们也可以根据自己的思考实现。

信号名	方向	用途	产生来源及机制
A1 [4:0]	I	读 CP0 寄存器编号	执行 `mfc0` 指令时产生

信号名	方向	用途	产生来源及机制
A2 [4:0]	I	写 CP0 寄存器编号	执行 `mtc0` 指令时产生
DIn [31:0]	I	CP0 寄存器的写入数据	执行 `mtc0` 指令时产生
PC [31:2]	I	中断/异常时的 PC	
ExcCode[6:2]	I	中断/异常的类型	异常功能部件
HWInt[5:0]	I	6 个设备中断	外部设备
WE	I	CP0 寄存器写使能	执行 `mtc0` 指令时产生
EXLSet	I	置位 SR 的EXL 位	流水线 W 级产生
EXLClr	I	置 0 SR 的EXL 位	执行 `eret` 指令时产生
clk	I	时钟信号	
rst	I	复位信号	
IntReq	O	中断请求	由 CP0 模块确认响应中断
EPC[31:2]	O	EXC 寄存器输出至 NPC	
DOut[31:0]	O	CP0 寄存器的输出数据	执行 `mfc0` 指令时产生

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。

届时我们也会在讨论区放出官方答疑贴，但我们仅会对我们认为有必要回答的问题进行回答（涉及测试的统一要求）。

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
6. .....