# CP0设计文档

异常处理单元，处理中断和系统调用。

以下描述中断均指外部中断。

# **寄存器格式：**

## STATUS寄存器[31:0] 编号12

记录系统调用号和中断掩码

记录当前cp0状态信息，主要包括中断信息，是否允许中断和系统调用。

第0位记录外部中断允许位，1表示允许中断，0表示屏蔽中断。

第8-15位标志是否允许具体的中断。

其他位不定义

初始化为32’hff01

## CAUSE寄存器[31:0] 编号13

第0为恒为0

第2-6位记录系统调用号。

第8-15位记录中断请求。

第8-9位对应status寄存器中相应位置，分配给软件中断

第10-15位对应status寄存器中相应位置，分配给硬件中断

其他位不定义

初始化为32’h0

## EPC寄存器[31:0] 编号14

记录中断返回地址, 在异常处理和发生中断时记录中断返回地址。

初始化为32’h0

## COUNT寄存器[31:0] 编号 9

时间计数器，流水线时钟信号的函数，以固定速度增加。达到32’hffffffff就清零

初始化为32’h0

# **输入：**

### Wb\_cp0\_w\_en

mtc0写使能，高电平有效，由id段产生，memwb提供。

### Cu\_cp0\_w\_en

cu的异常写使能，高电平有效，cu提供。

### Epc [31:0]

记录中断返回地址，cu提供。

### Id\_cp0\_src\_addr [4:0]

cp0源寄存器地址，id译码出来直接送入，读出来的数据在id段使用， id译码单元提供。mcf0指令使用。

### Wb\_cp0\_dst\_addr [4:0]

cp0寄存器地址，memwb提供，mtc0指令使用。

### Ex\_data [31:0]

来自ex段的寄存器数据，memwb提供，是即将要写入cp0[wb\_cp0\_dst\_addr]的数据。

### Cu\_exec\_code [4:0]

来自cu的异常号。写入cause[6:2]

### Clk

时钟信号

# **输出**

### Cp0\_data [31:0]

mfc0指令请求的数据，提供给ID

### Cp0\_epc [31:0]

中断返回地址，epc寄存器的值。提供给cu

### Cp0\_intr

组合逻辑，中断信号。提供给cu

# **cp0行为：**

## 时钟计数器增加

每次有效脉冲，count寄存器加一

## 检测到中断

中断存在的定义如下：

1. status[i]值为1(8<= i <=15)
2. cause[i]值为1
3. status[0]值为1

以上三个条件同时成立即产生中断。

检测到中断之后，从cp0\_intr端口将中断输出。并填写cause寄存器。

Cu在拿到cp0\_intr的信号之后，cu\_cp0\_w\_en有效送回cp0。cu\_exec\_code[4:0]的值为0。

## MTC0

wb\_cp0\_w\_en使能时，将ex\_data数据写入cp0[wb\_cp0\_dst\_addr]。Status寄存器的编号为12，cause寄存器的编号为13，epc寄存器编号为14， count寄存器编号为9。

## MFC0

将cp0[wb\_cp0\_src\_addr]数据从cp0\_data输出，编号同MTC0。

## 捕获到系统调用或中断

当cu\_cp0\_w\_en使能时，捕获系统调用或中断，将cu\_exec\_code[4:0]的值写入cause[6:2]，提供给上层软件处理。保存epc，并对status[0]置0来屏蔽中断。

## 异常/中断返回

交给软件实现

上述行为除时钟计数器外优先级由低到高。

中断期间，cu不应该产生系统调用

如果在MTC0信号之后三个时钟内产生MFC0指令，并且写入和读取的是同一个cp0寄存器，则发生cp0冒险，实验忽视这个冒险，交给编译器去避免。