关于CLIC的调研报告

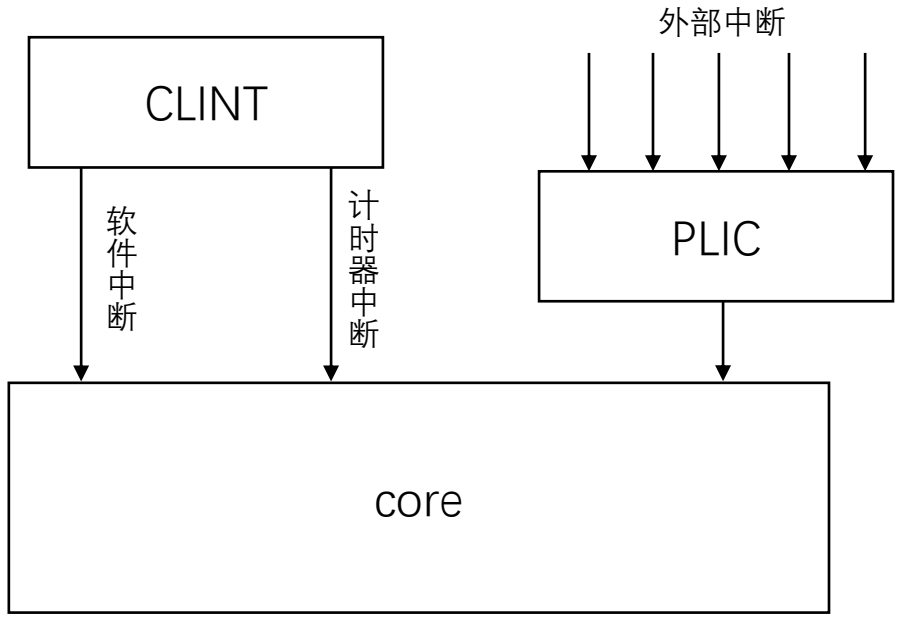
——Core-Local Interrupt Controller

PLCT实验室：史玉龙
Email：shiyulong@iscas.as.cn

- CLIC是什么？
- CLIC概述
- CLIC中断处理

CLIC是什么？

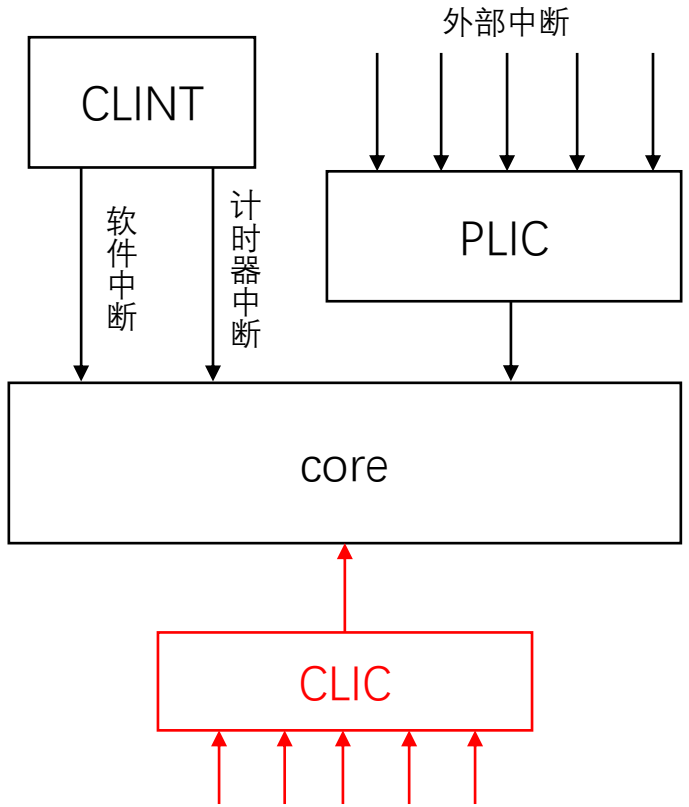
- riscv现有的中断及处理模块



- risc-v中断主要分为局部中断和全局中断；
- 局部中断即本地中断，包括软件中断和计时器中断，由CLINT处理
- 外部中断即全局中断，由PLIC进行处理，仲裁出一个中断源给核
- 优先级：外部中断>软件中断>定时器中断

CLIC是什么？

- CLIC(Core-Local Interrupt Controller)



- CLIC：内核本地中断控制器，接收中断信号并提供要由Hart处理的下一个中断，其宗旨在于为risc-v系统提供低时延、向量化、抢占式中断
- 目标在于为各种软件ABI和中断模式提供支持
- 激活CLIC包含并替换原始的RISC-V基本本地中断机制
- CLINT基于早期的设计提供本地中断，管理中断源，且也支持在xip寄存器的第16位及以上添加定制的快速中断信号
- PLIC为共享的平台级中断提供集中的优先级和路由，为每一个hart发送一个外部中断信号

CLIC概述

- 每个hart的CLIC支持最多4096个中断输入，每个中断输入有4个8位寄存器保存中断源信息，分别为clicintip[i]、clicintie[i]、clicintattr[i]、clicintctl[i]
- 前16个中断输入保留，所以支持最多4080个local external interrupts被添加
- 通过将中断划分为256个级别和优先级来支持中断抢占机制

CLIC概述

- CLIC Memory Map(M-mode):

地址范围	读写权限	寄存器
0x0000	RW	cliccfg
0x0004	R	clicinfo
0x0040-0x00BC	RW	clicinttrig[i] (0≤i≤31)
0x1000+4*i-0x4FFC	R or RW	clicintip[i] (0≤i≤4095)
0x1001+4*i-0x4FFD	RW	clicintie[i] (0≤i≤4095)
0x1002+4*i-0x4FFE	RW	clicintattr[i] (0≤i≤4095)
0x1003+4*i-0x4FFF	RW	clicintctl[i] (0≤i≤4095)

CLIC概述

- cliccfg寄存器：其定义了支持哪些特权模式、clicintctl[i]寄存器如何划分中断级别和优先级，以及是否支持选择硬件向量。字段如下：

bits	field	description
7	reserved(0)	当前规范下缺省为0
6:5	nmbits[1:0]	设置clicintattr[i].mode字段有多少个bit的物理实现
4:1	nlbits[3:0]	表示clicintctl[i]寄存器分配了多少个高位来编码中断级别(当前只支持0-8个)，剩下的比特位的最低有效位作为优先级有效位
0	nvbits	是否实现了选择性中断硬件向量功能(0表示不选择)

CLIC概述

- clicinfo寄存器：一个只读寄存器，显示一些对调试有用的信息。字段如下：

bits	field	description
31	reserved(0)	当前规范下缺省为0
30:25	num_trigger	指定实现中支持的最大中断触发器数，有效值为0-32
24:21	CLICINTCTLBITS	指示clicintctl寄存器中实际实现了多少硬件比特位
20:13	version	指示CLIC的实现版本
12:0	num_interrupt	指示实现时能够支持的最大中断输入的实际数量

CLIC概述

- clicintip寄存器
- 规范为每一个中断输入都分配了专用的挂起比特位，这个比特位就是clicintip寄存器的第0位。clicintip[i]=0时，无中断挂起；clicintip[i]!=0存在中断挂起。

CLIC概述

- clicintie寄存器
- 规范为每一个中断输入都分配了专用的中断使能比特位，通过对该位的读写，来控制中断的启动和禁用。该使能位为clicintie寄存器的第0位。 clicintie[i]=0时，无中断使能， clicintie[i]!=0表示中断使能。

CLIC概述

- clicintattr寄存器：一个读写寄存器，用于为每个中断指定其各种属性。字段如下：

bits	Field	description
7:6	mode	指示该中断在哪个模式下运行,与mstatus.mpp编码相同。11:M-mode;01:S-mode;00:U-mode
5:3	reserved(0)	当前规范下缺省为0
2:1	trig	指示中断输入的触发类型和极性。trig[0]定义触发方式(0-电平触发，1-边沿触发)；trig[1]定义极性(0-上升/高电平，1-下降/低电平)
0	shv	用于选择性硬件向量化，0分配为非向量中断；1分配为硬件向量中断. cliccfg.nvbits为0时，该位硬件连线为0.

CLIC概述

- clicintctl寄存器：一个读写寄存器，用于为每个中断指定其中断级别和中断优先级。

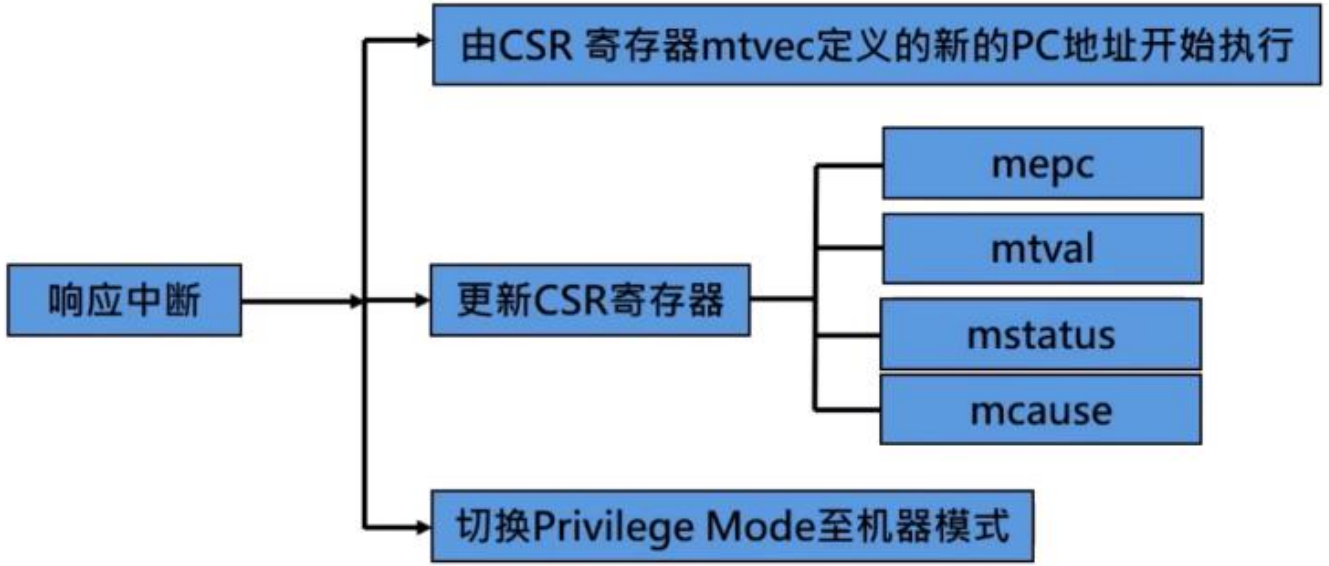
CLIC概述

- clicinttrig寄存器：此寄存器为可选寄存器，用于生成断点异常、进入debug模式或跟踪。实际支持的触发器数量在clicinfo.num_trigger字段中指定。字段如下：

bits	field	description
31	enable	控制中断的启用/禁用
30:13	reserved	设置clicintattr[i].mode字段有多少个bit的物理实现
12:0	Interrupt_number	选择哪一个中断输入号作为该中断触发的源

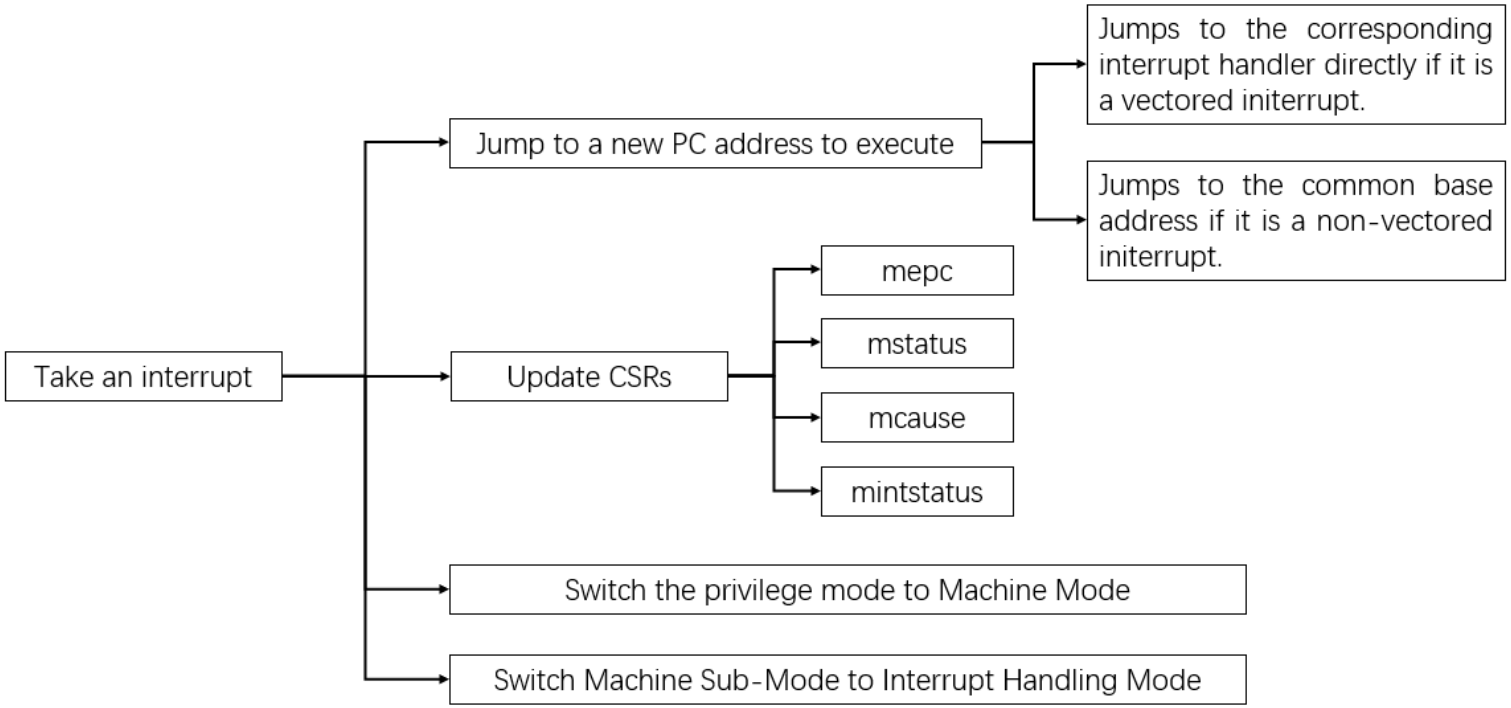
CLIC中断处理

- 现有中断模式下，中断发生时行为鉴赏[1]：



CLIC中断处理

- CLIC模式下，中断发生时行为鉴赏[2]：



CLIC中断处理

- 在CLIC模式下，寄存器行为的变化(spec第5章)：

CSR	Description
mstatus	用于存储中断或异常处理模式状态
medeleg	用于将异常重定向对应的特权模式下处理
mideleg	用于将中断重定向对应的特权模式下处理
mie	用于屏蔽非CLIC模式下的对应中断
mtvec	用于存储异常或者中断入口地址
*mtvt	用于存储CLIC模式向量中断的基地址
mscratch	用于机器模式下的程序临时保存某些数据
mepc	用于存储异常或者中断的返回地址
mcause	用于存储异常或中断原因
mtval	用于存储导致出现异常的地址或者指令
mip	用于查询中断等待状态

CLIC中断处理

- 续：

CSR	Description
*mnxti	用于获取中断入口地址和修改全局中断使能
*mintstatus	用于查询各模式下的对应中断的level
*mscratchcsw	用于特权模式切换时的数据交换
*mscratchcsw1	用于中断level切换时的数据交换

- 带*的寄存器表示引入CLIC后新设计的寄存器
- 中断发生时，刷新mepc、mstatus、mcause、mintstatus寄存器

CLIC中断处理

现有模式下的mtvec寄存器

bits	field	description
31:2	base	异常的基地址
1:0	mode	异常模式

- mode=0时，pc跳转到base地址执行
- mode=1时，同步异常pc跳转到base地址执行；异步中断则跳转到base+4*exccode执行
- 要求基地址为4字节对齐
- 原始的基本向量模式只是跳转到向量表中的一个地址

CLIC模式下的mtvec寄存器

bits	field	description
31:6	base	异常的基地址
5:0	mode	异常模式

- Mode最低2个有效位为11，要求基地址为64字节或更大的二次幂对齐
- 非向量中断，直接跳转到陷阱处理程序地址执行；
- 向量中断，PC值从mtvt中获取。
- CLIC模式则是从表中读取一个处理函数地址，直接在硬件中跳转到该地址

CLIC中断处理

- mtvt寄存器——用于存放CLIC向量中断处理函数基地址

```
pc := M[TBASE + XLEN/8 * exccode] & ~1
```

```
         # Vector table layout for RV32 (4-byte function pointers)
mtvt ->  0x800000 # Interrupt 0 handler function pointer
         0x800004 # Interrupt 1 handler function pointer
         0x800008 # Interrupt 2 handler function pointer
         0x80000c # Interrupt 3 handler function pointer

         # Vector table layout for RV64 (8-byte function pointers)
mtvt ->  0x800000 # Interrupt 0 handler function pointer
         0x800008 # Interrupt 1 handler function pointer
         0x800010 # Interrupt 2 handler function pointer
         0x800018 # Interrupt 3 handler function pointer
```

CLIC中断处理

现有模式下的mcause寄存器

bits	field	description
31	interrupt	当trap由一个中断造成时，对应的中断位置1
30:0	exception code	记录上次异常的原因代码 WLRL域

- 中断发生时，记录中断的原因
- exccode用于参与计算入口地址

CLIC模式下的mcause寄存器

bits	field	description
31	interrupt	当trap由一个中断造成时，对应的中断位置1
30	minhv	Set by hardware at start of hardware vectoring, cleared by hardware at end of successful hardware vectoring
29:28	mpp	记录前一个中断的模式，与mstatus.mpp相同
27	mpie	记录前一个中断的使能位，与mstatus.mpie相同
26:24	reserved	
23:16	mpil	记录上次中断的level
15:12	reserved	
11:0	exception code	记录上次异常的原因代码

CLIC中断处理

bits	field	description
31	interrupt	当trap由一个中断造成时，对应的中断位置1
30	minhv	Set by hardware at start of hardware vectoring, cleared by hardware at end of successful hardware vectoring
29:28	mpp	记录前一个中断的模式，与mstatus.mpp相同
27	mpie	记录前一个中断的使能位，与mstatus.mpie相同
26:24	reserved	
23:16	mpil	记录上次中断的level
15:12	reserved	
11:0	exception code	记录上次异常的原因代码

- CLIC模式则扩展mcause寄存器，以记录更多有关中断上下文的信息，用于减少为mret指令保存和恢复该上下文的开销。
- mpp和mpie和mstatus寄存器中对应的字段相同，可以减少上下文保存和恢复的代码

CLIC中断处理

- mnxti寄存器——用于获取中断入口地址和修改全局中断使能
- 使用csrrsi/csrrci指令访问
- 相同特权模式下，中断的level高于保存的中断上下文中的level且高于该特权模式的中断阈值时，软件可以利用mnxti寄存器服务下一个中断，且不会产生中断流水线刷新和上下文保存/恢复的全部成本。

参考资料

- [1]友晶科技关于RISC-V on T-Core的处理器中断与异常：https://www.bilibili.com/video/av541622331
- 芯来科技关于蜂鸟E203异常与中断：https://www.bilibili.com/video/BV1H5411t7Ct/?spm_id_from=333.788.recommend_more_video.-1
- CLIC spec:https://github.com/riscv/riscv-fast-interrupt.git
- Nuclei RV-STAR开发板之快速中断入门： https://www.riscv-mcu.com/site/nuclei_interrupt_quickstart/
- [2]Nuclei ISA Spec： https://doc.nucleisys.com/nuclei_spec/index.html
- CSDN相关系列帖子