

中断异常

高泽宇

中国科学院大学

2021/8/15



上海处理器技术创新中心
SHANGHAI INNOVATION CENTER FOR COMPUTING TECHNOLOGY



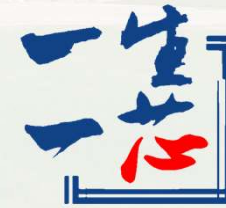
鹏城实验室
Peng Cheng Laboratory



中国科学院大学
University of Chinese Academy of Sciences



北京智源
3A AI



ByteDance
字节跳动



中国开放指令生态(RISC-V)联盟
China RISC-V Alliance



中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

■ 提纲

- 中断异常简介
- 实现环境调用异常
- 实现计时器中断
- 小结



第一部分

中断异常简介



什么是中断和异常

- 异常(狭义): 一般指来源于核内部的异常
 - 如: 指令错误、访存错误等
- 中断: 一般指来源于核外部的异常, 属于广义的异常
 - 包括外部中断、计时器中断以及软件中断
- trap: 由异常或中断引起的控制转移

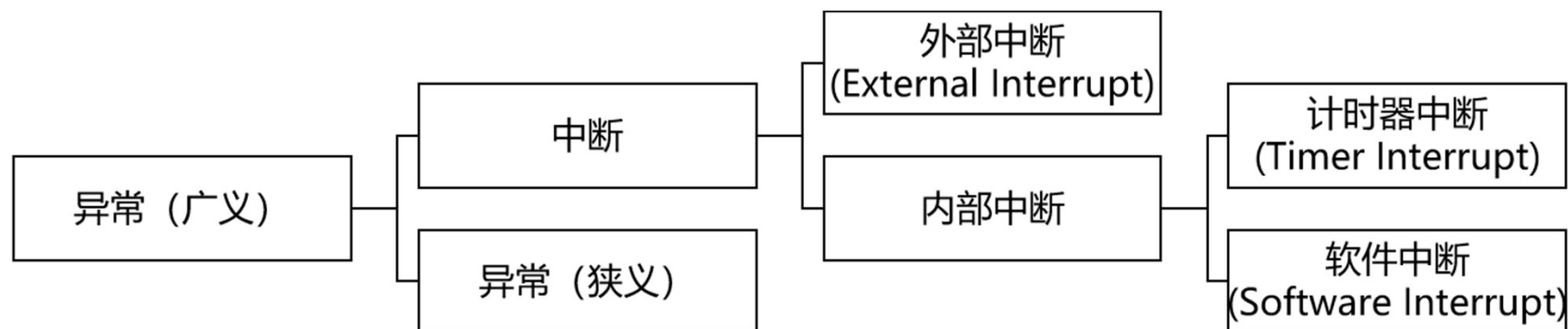


什么是异常处理机制

- 处理器内核在执行程序的过程中，突然发生异常，中止当前执行的程序，转而去做别的事情。
- 当别的事情处理完后，回到之前的地方继续执行之前的程序。



RISC-V异常类型



- 要运行interrupt test (am-kernels/tests/am-tests), 暂时只需要实现时钟中断+环境调用异常



特权级别

- 在任何时候，RISC-V处理器一定是运行在一个特权级别当中的

Level	Encoding	Name	Abbreviation
0	00	User/Application	U
1	01	Supervisor	S
2	10	<i>Reserved</i>	
3	11	Machine	M

Table 1.1: RISC-V privilege levels.

- 暂时只需要实现M模式

注意事项

- 中断嵌套
 - 处理器内核正在处理中断时，出现了新的中断请求
 - RISC-V 在硬件层面不支持，需要软件层面的特殊处理
 - 所以暂时不需要考虑
- 仲裁
 - 当同时出现多个中断的时候，需要进行仲裁
 - 出现在外部中断与内部中断、外部与外部中断之间
 - 所以暂时不需要考虑
- 中断处理模式
 - 分为向量处理模式和非向量处理模式
 - 暂时只需要实现非向量处理模式



第二部分

实现环境调用异常



需要的CSRs

- 控制状态寄存器 **C**ontrol and **S**tatus **R**egisters

Number	Privilege	Name	Description
Machine Information Registers			
0xF11	MRO	mvendorid	Vendor ID.
0xF12	MRO	marchid	Architecture ID.
0xF13	MRO	mimpid	Implementation ID.
0xF14	MRO	mhartid	Hardware thread ID.
Machine Trap Setup			
0x300	MRW	mstatus	Machine status register.
0x301	MRW	misa	ISA and extensions
0x302	MRW	medeleg	Machine exception delegation register.
0x303	MRW	mideleg	Machine interrupt delegation register.
0x304	MRW	mie	Machine interrupt-enable register.
0x305	MRW	mtvec	Machine trap-handler base address.
0x306	MRW	mcounteren	Machine counter enable.
Machine Trap Handling			
0x340	MRW	mscratch	Scratch register for machine trap handlers.
0x341	MRW	mepc	Machine exception program counter.
0x342	MRW	mcause	Machine trap cause.
0x343	MRW	mtval	Machine bad address or instruction.
0x344	MRW	mip	Machine interrupt pending.

[1]The RISC-V Instruction Set Manual Volume II: Privileged Architecture Document Version 20190608-Priv-MSU-Ratified

mstatus

- Machine-mode Status Register
- 跟踪并控制处理器的当前运行状态
- 可读/写

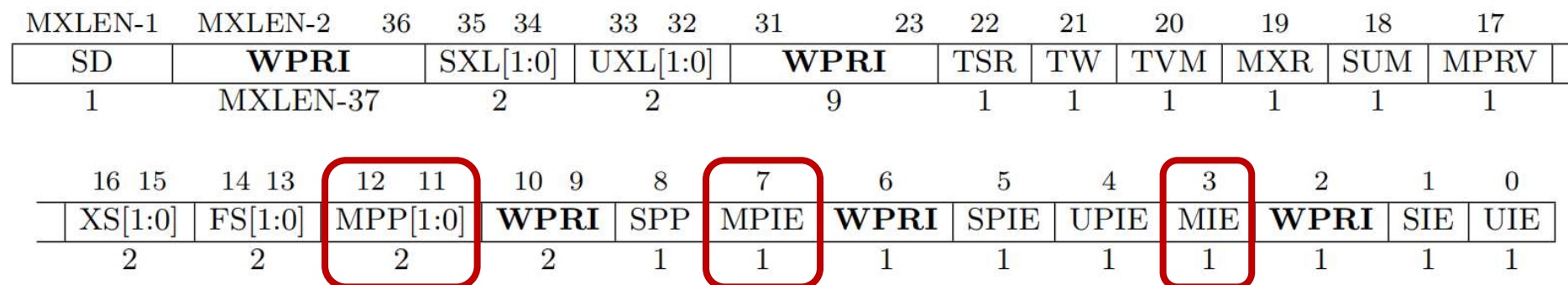


Figure 3.7: Machine-mode status register (mstatus) for RV64.

mstatus-MIE

- 启用/禁用全局中断
- 当硬件线程在特权模式 M 下执行时
 - MIE=1: 全局启用中断
 - MIE=0: 全局禁用中断

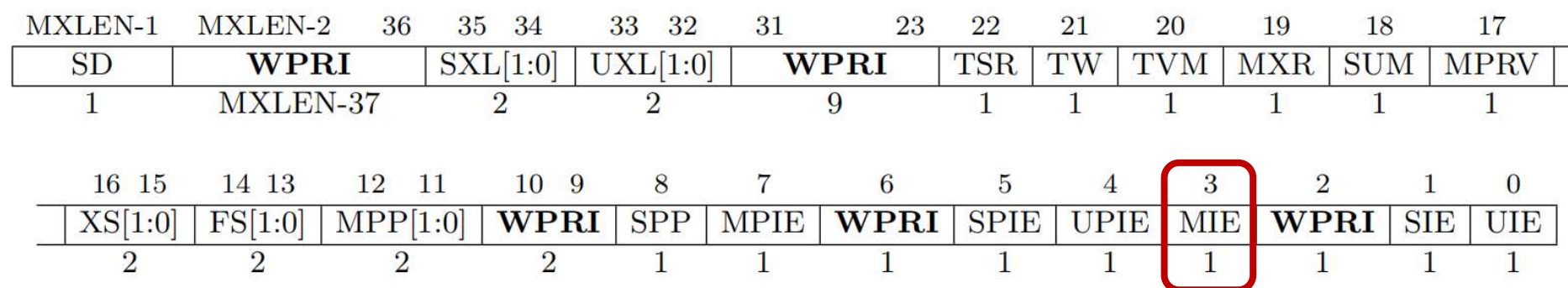


Figure 3.7: Machine-mode status register (mstatus) for RV64.

mstatus-MPIE & MPP

- 为了中断结束后能恢复原本的MIE位和特权模式
- MPIE: 在 trap 之前 MIE 位的值
- MPP : 在 trap 之前的特权模式 (M->3)

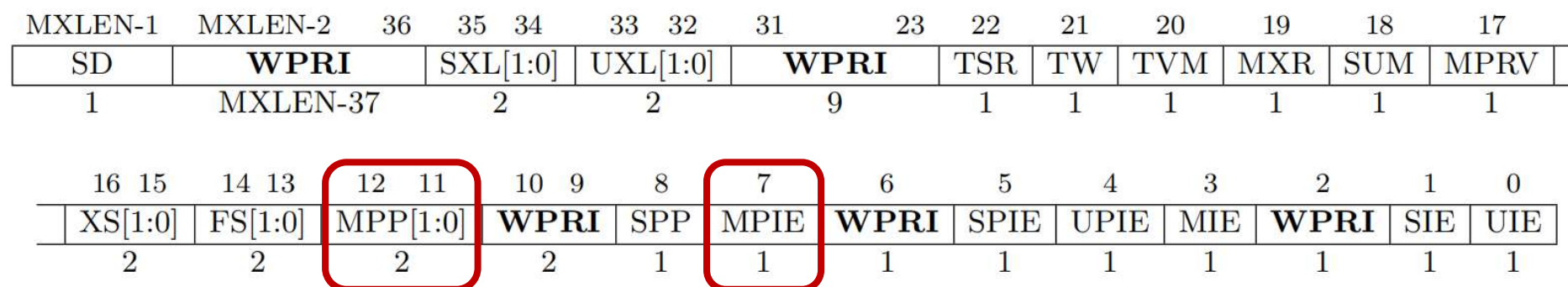


Figure 3.7: Machine-mode status register (mstatus) for RV64.

mtvec

- **M**achine **T**rap-**V**ector Base-Address Register
- trap 向量配置 (异常入口地址)
- 可读/写

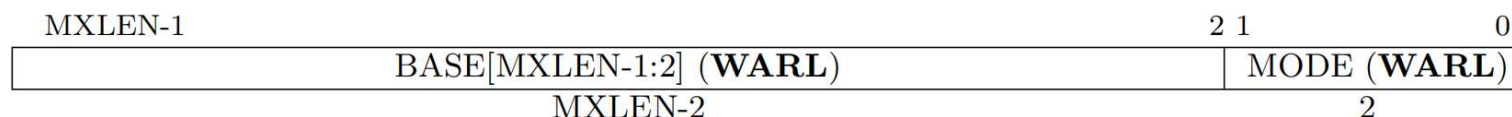


Figure 3.8: Machine trap-vector base-address register (mtvec).

Value	Name	Description
0	Direct	All exceptions set pc to BASE.
1	Vectored	Asynchronous interrupts set pc to BASE+4×cause.
≥2	—	Reserved

Table 3.5: Encoding of mtvec MODE field.



mepc

- Machine Exception Program Counter
- 遇到中断/异常的指令的 pc
- 可读/写

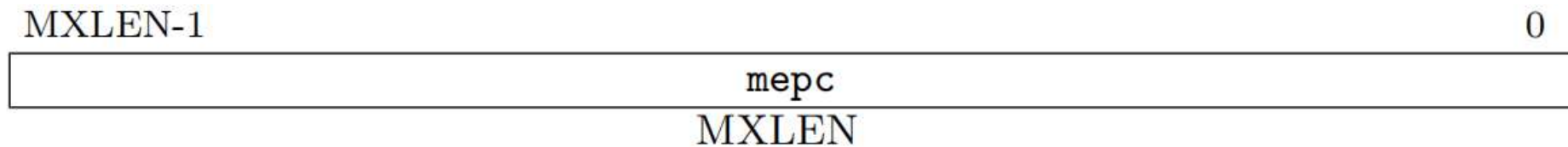


Figure 3.21: Machine exception program counter register.



mcause

- Machine Cause Register
- 指示导致 trap 的事件
- 可读/写
- 如果 trap 是由中断引起的，中断位则会置1。

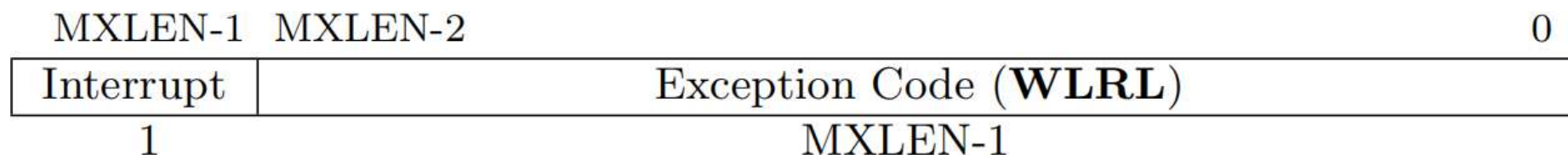


Figure 3.22: Machine Cause register mcause.



编写相关代码-软件

- 编写异常处理代码
- 设置异常处理入口地址
- 调用ecall指令



进入异常-硬件

- CPU捕获异常
- 保存/更新CSRs
 - mepc
 - 异常：当前的指令的 PC 值 (**ecall**)
 - mcause
 - 记录当前的异常原因： 11
 - mstatus
 - MPIE设为MIE的值，MIE设为0
- 跳转到mtvec的异常入口地址



进行异常处理-软件

- 保存上下文：保存CPU的寄存器到内存中
- 跳转到异常处理程序进行异常处理（ $pc + 4$ ）
- 恢复上下文：从内存中恢复CPU的寄存器
- PS：RISC-V架构的处理器不实现硬件上的上下文保存/恢复，需要在软件上实现上下文的切换机制



退出异常(mret)-软件

- 恢复mstatus
 - MIE从MPIE中恢复, MPIE设置为1
- 跳转到mepc中的地址



第三部分

实现计时器中断



mip

- Machine interrupt-pending register
- 包含等待中断的信息
- 可读/写*

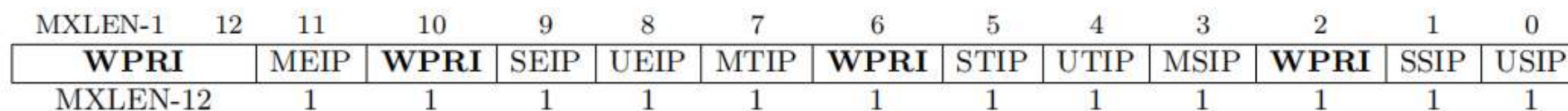


Figure 3.11: Machine interrupt-pending register (mip).



mie

- **M**achine **i**nterrupt-**e**nable register
- 包含中断启用位
- 可读/写
- MTIE: M模式计时器中断启用位

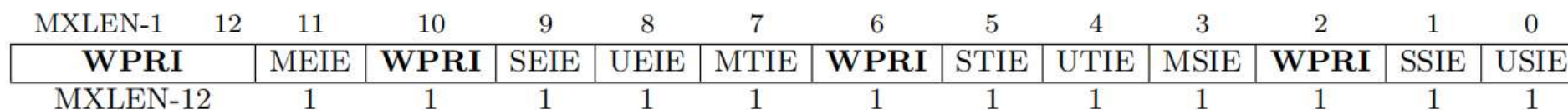


Figure 3.12: Machine interrupt-enable register (mie).



mtime(Clint)

- Machine time register
- 64位内存映射的M模式实时计数器
- 不是CSR, 可读/写
- 属于Clint (Core-Local Interruptor), 外设
- 以恒定的频率增加

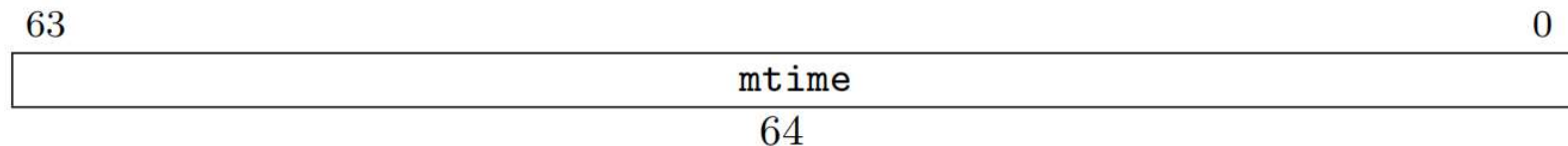


Figure 3.13: Machine time register (memory-mapped control register).



mtimecmp(Clint)

- Machine time compare register
- 64位内存映射的 M 模式计时器比较寄存器
- 不是CSR
- 可读/写
- 当 $\text{mtime} \geq \text{mtimecmp}$ 时, 产生计时器中断, mip的MTIP位置1

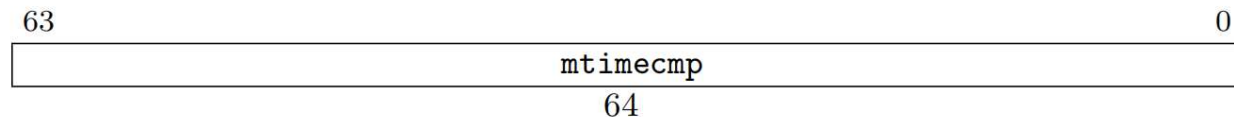


Figure 3.14: Machine time compare register (memory-mapped control register).



编写相关代码-软件

- 编写中断处理代码（打印t）
- 在启动中设置CSR
- 开中断
 - mstatus的MIE位
 - mie的MTIE位



进入中断-硬件

- CPU捕获计时器中断
- 保存/更新CSRs
 - mepc
 - 中断：下一条指令的PC值
 - mcause
 - 记录当前的异常原因：0x8000000000000007
 - mstatus
 - MPIE设为MIE的值，MIE设为0
- 跳转到mtvec的异常入口地址



进行中断处理-软件

- 保存上下文：保存 CPU 的寄存器到内存中
- 跳转到中断处理程序进行中断处理（**mtimecmp**）
- 恢复上下文：从内存中恢复 CPU 的寄存器



退出异常(mret)

- 恢复 mstatus
 - MIE 从 MPIE 中恢复, MPIE 设置为 1
- 跳转到 mepc 中的地址

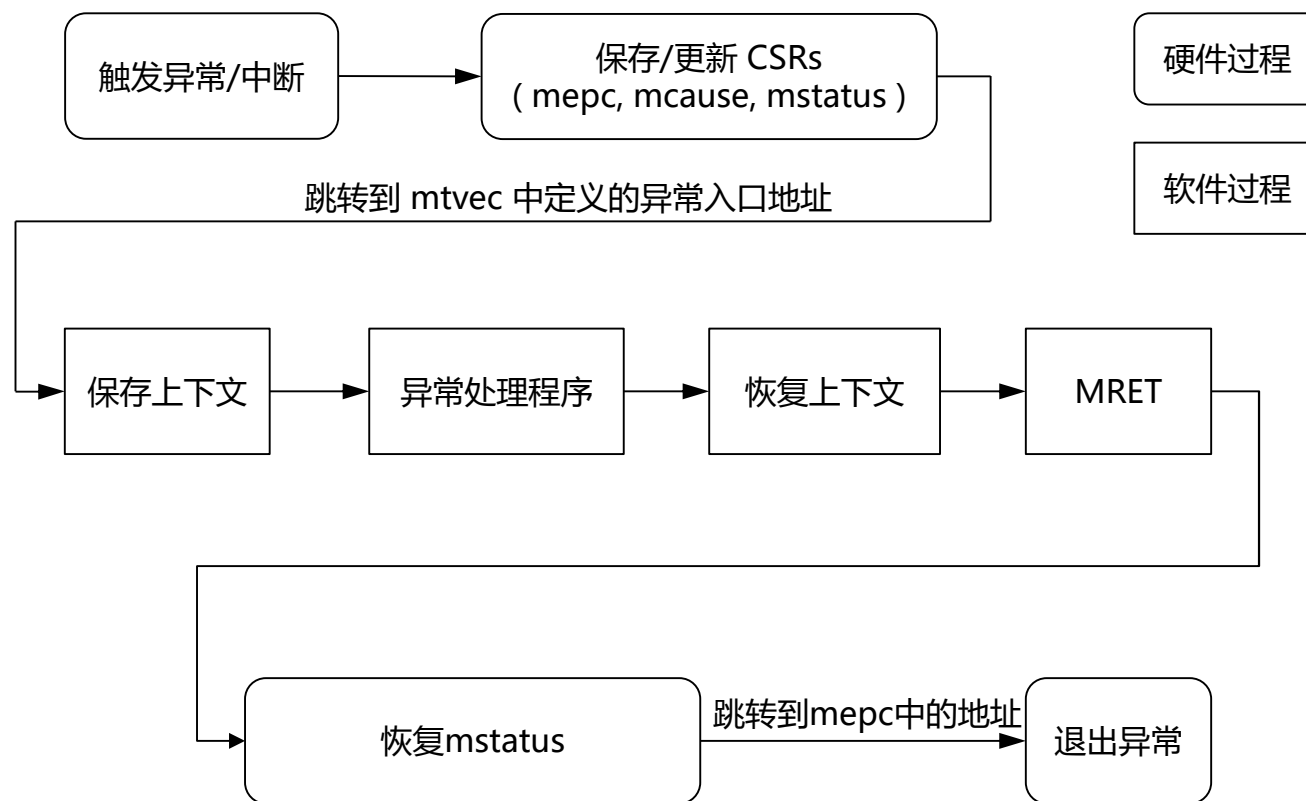


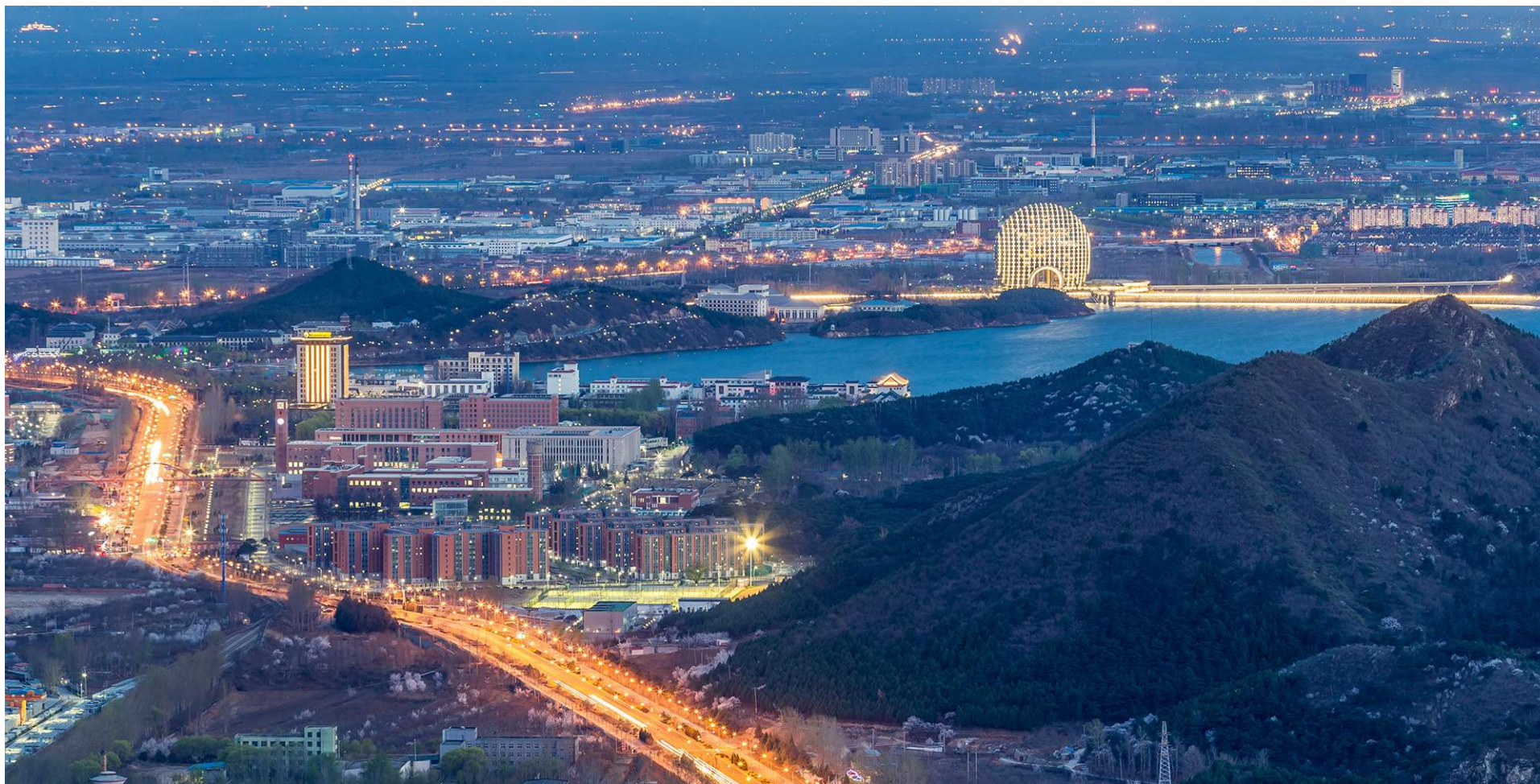
第四部分

中断异常的全流程



小结





谢谢，欢迎批评指正！



中国科学院大学
University of Chinese Academy of Sciences