RISC-V SBI用例

1. 基于SBI实现对S模式指令集的扩展。

应用场景:

1. CPU可能没有实现某条指令对应的处理器扩展，在CPU算力足够的情况下，可以利用非法指令陷入的方式，使处理器陷入M模式，由SBI对相应指令进行模拟实现，再将结果返回给S模式继续处理，整个过程S模式不感知。
2. S模式的OS扩展自定义指令来简化某些常用操作，通过内嵌汇编的方式调用扩展指令，实现对指定操作的定义.

举例来说，系统想通过定义扩展指令的方式来简化矩阵乘法的操作，可以定义指令:

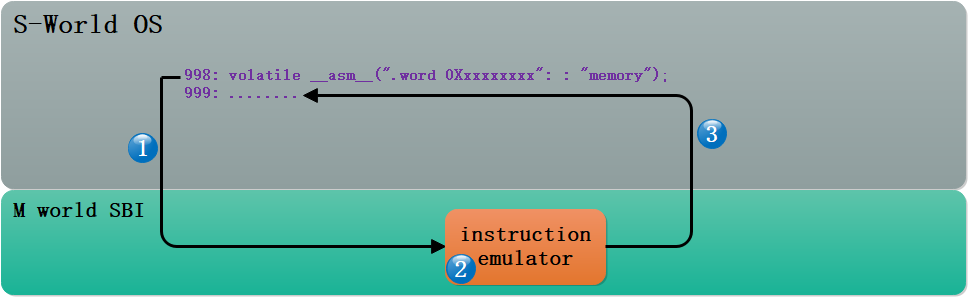
**Matrixply dst, src1, src2**

其中,dst,src1, src2均为ISA定义内GPR，Src1指向 m\*n维矩阵buffer, src2指向n\*p维矩阵buffer, dst

指向m\*p维的运算结果buffer.

软件流程如下图所示：

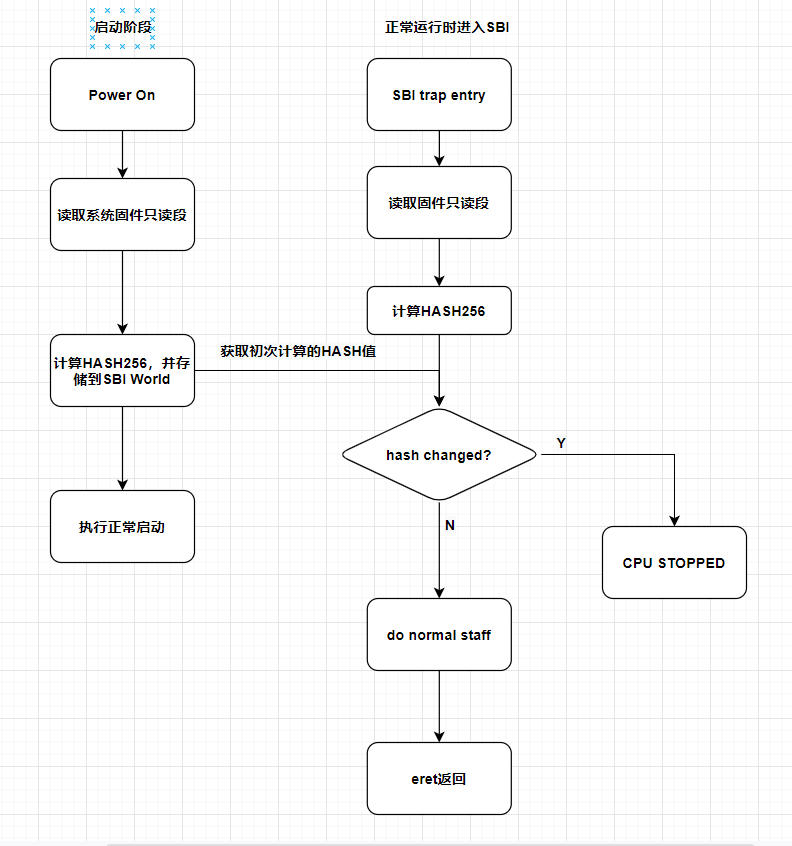
1. S模式触发非法指令异常，该异常被delegate到M模式处理
2. 进入到M模式异常控制流执行，执行流中获取指令信息，src,dst寄存器信息等，在emulator程序中完 成计算，将计算结果写入dst.
3. 异常执行完，执行异常返回指令，回到OS的下一行继续执行：



1. 编写SBI固件扩展，实现对S模式固件的完整性检测,当检测到S模式运行的程序被修改，则CPU在M模式stop,不返回S模式.

应用场景：安全场景，要求OS系统在运行中没有发生修改。

程序流程:



3:编写SBI固件，支持系统调用重定向扩展，当S模式通过ecall指请求系统服务的时候，要求：

1. :对于SBI支持的系统调用，由SBI实现处理，比如SBI legcay的服务等等。
2. 对于SBI没有实现或者无法实现的系统调用，要求SBI将其delegate到S模式，由操作系统处理，不能返回不支持。

此项功能对于需要划分CPU模式的大型RTOS，或者对于一部分允许通过ecall 指令申请S模式系统服务的模块化OS比较重要。

程序流程：

