RISC-V ULI: Rocket Chip support

Gwyneth Chen

October 2022

1 Enabling ULIs on Rocket Chip

As it stands, user-level interrupts are still only a proposed extension and are therefore not supported by Rocket Chip. Unless otherwise specified, all code discussed in the following section exists in the file `src/main/scala/rocket/CSR.scala`.

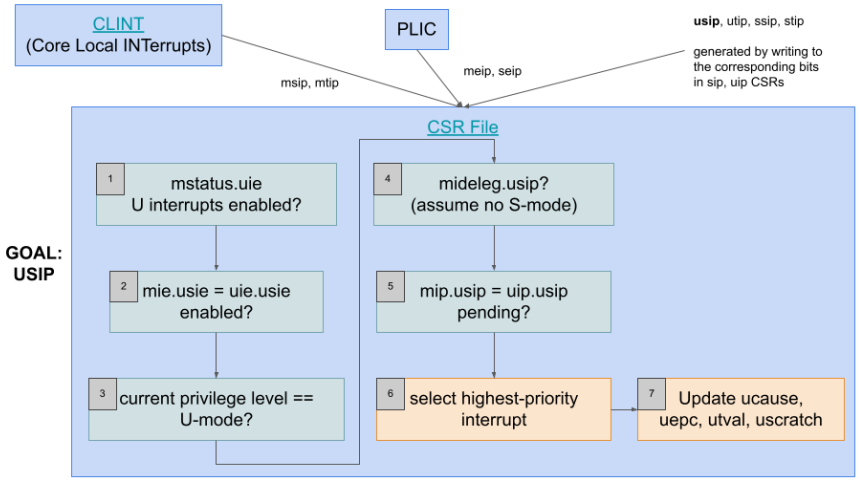

Figure 1: Target USIP processing on Rocket Chip

Figure 1 illustrates the rough pathway that we wish a pending user software interrupt (USIP) to travel, defined analogously to the current handling for machine and supervisor-level interrupts. The USIP is first created by writing the corresponding interrupt-pending bit in the UIP register, which is a view of the MIP register with only the user interrupt bits visible in U-mode. In the CSR file (defined as object CSRFile), a number of checks (1-5 in the diagram) are performed to generate the mask of interrupts that are both pending and viable. For a USIP to be handled in U-mode, we want these checks include (1)

whether user-level interrupts are enabled at all, (2) whether user software interrupts are specifically enabled, (3) whether the current privilege level is U-mode, (4) whether USIPs have been delegated to U-mode, and of course, (5) whether a USIP is actually pending. Once the mask of possible interrupts has been created, the highest-priority one is then selected, and the corresponding CSR states are updated. Note that for check (4), we are currently only checking `mideleg`, which indicates which interrupts are delegated from M-mode. In a system where supervisor mode (S-mode) also exists, we would need to check an additional `sideleg` register; however, this register is not yet defined in Rocket Chip, so we are assuming no S-mode exists in order to simplify this first step.

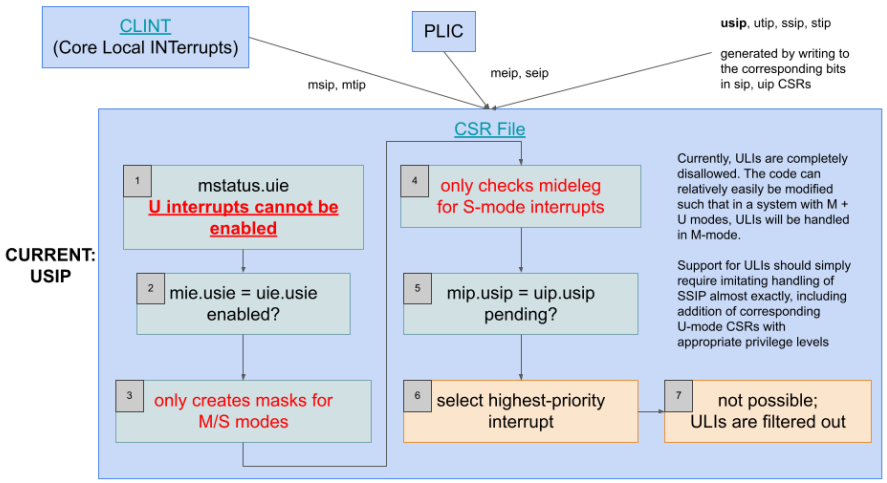

Figure 2: Current USIP processing on Rocket Chip

In contrast, Figure 2 illustrates the current situation; i.e., those parts that have been modified in the workflow.

1. Since user interrupts are not yet supported, `mstatus.uie` is in fact always set to 0. We instead wish to change this so that it is initialized to 1 when U-mode exists in the system.

   Note that changing only this step should be sufficient to allow USIPs to be enabled for handling in higher privilege modes, i.e. handled in M-mode if one is triggered while running in U-mode.

3. We need to modify the existing interrupt masking to consider handling user-level interrupts when we are running in U-mode.

4. `mideleg` should be checked at the bit corresponding to USIP

7. The U-mode CSRs for interrupt handling must be added to the CSR file analogously to their S- and M-mode counterparts (uepc, utvec, etc.)