PLIC

External Interrupts:

- Interrupts are asynchronous events generated by a external source through hardware.The processor services the interrupts. In RISC-V interrupts are classified into timer,software and external interrupts. The external interrupts are also called as global interrupts. Timer and software interrupts are handled by a Core Local Interrupt (CLINT). External interrupts are handled by the PLIC.

Understanding PLIC working

- The PLIC connects the global interrupt sources to the interrupt target i.e., core. The PLIC consists of the ”PLIC core” and the ”Interrupt gateways”. There are multiple interrupt gateways, one per interrupt source. Global interrupts are sent from theirsource to one of the interrupt gateway.
- The interrupt gateway processes the arriving interrupt signal from each source and sends a single interrupt request to the PLIC core. The PLIC core contains a set of interrupt enable (IE) bits to enable individual interrupt sources in the PLIC. The PLIC core contains pending interrupt bits to signal that an interrupt is waiting to be processed.
- Also, PLIC core performs interrupt prioritization/arbitration. Each interrupt source is assigned a separate priority. The PLIC core latches the interrupt request into the Interrupt Pending bits (IP).
- Whenever, the priority of the pending interrupt exceeds a per-target threshold the PLIC core forwards an interrupt notification to the interrupt target. The PLIC Claim/Complete register holds the highest priority interrupt waiting to be processed.

1. The interrupt is taken by the core (Hart1) and handed over to the software application.The software application has a PLIC interrupt handler to service the interrupts.Before the interrupt is received by the software application, the core copies the value of mstatus.MIE into mstatus.MPIE, and then mstatus.MIE bit is cleared.
2. This disables any new interrupts. The privilege mode prior to the interrupt is set mstatus.MPPbit.
3. The privilege mode is set to Machine mode. The current value of PC is copied to MEPC register. Then, the core sets the Program Counter (PC) to point to ”mtvec” base address [1]. The base address holds the Interrupt handler routine.
4. Once the interrupt is serviced, the core sends the associated interrupt gateway, an interruptcompletion message. The interrupt completion message usually writes the interrupt id to the PLIC Claim/Complete register.
5. On, interrupt completion, the saved context is restored by the software application. Usually the core sets the privilege mode to the value encoded in mstatus.MPP.
6. And the PC is set to the value of mepc. The value of mstatus.MPIE bit is copied into mstatus.MIE bit. This essentially enables all the interrupts.
7. The interrupt gateway can now forward another interrupt request to the PLIC.