Interrupt implementation summary

This document summarizes the interrupt-related enhancements and supporting fixes for the ‘riscv_mic

Interrupt sources and priority
- Added three level-sensitive sources: external (‘INTERRUPT‘), timer (‘TIMER_INTERRUPT‘), and sof
- Requests are masked by ‘mstatus.mie‘ plus the corresponding ‘mie‘ bits, then prioritized external ove
- On a taken interrupt, the core saves the faulting PC to ‘mepc‘, vectors to address ‘0x00000080‘, and

CSR support
- Implemented CSR read/modify/write decoding for CSRRW, CSRRS, and CSRRC, allowing register c
- Machine CSRs include ‘mstatus‘ (global MIE), ‘mie‘ (per-source enables), ‘mepc‘ (return PC), and ‘m
- Interrupt acceptance is gated by these CSRs, aligning the external, timer, and software sources with

Testbench coverage
- The ‘riscv_micro_interrupt_tb‘ testbench pulses simultaneous external and timer interrupts after reset
- Fetch addresses are logged for debug visibility, and the run window was extended to allow sufficient

Supporting microarchitectural fixes
- Sequencer reset was tightened so the state machine deterministically enters the fetch state after res
- Control-word clearing was deferred until the load/store clear phase, allowing the ‘load_store_unit‘ to i