Section	Title	Description	Link	Type	Weight	Goal
1	PLIC Internal Register Config	Following registers are to be configured, 1. Interrupt Enable (32'h80040088) 2. Interrupt Threshold (32'h80040090) 3. Interrupt Priority (32'h80040004)				
1.1	Interrupt Enable	Interrupt Enable handles logic controlling which registers are masked. Each interrupt source has a bit corresponding to it in the Interrupt Enable (mie) register. Checker : Check if the interrupt resolution winner and the interrupt sources latched in mie are disabled/masked or not				
1.2	Interrupt Threshold	Inerrupt sources with priority below threshold will be masked. Checker : Check if the interrupt resolution winner and the interrupt sources latched are above threshold				
1.3	Interrupt Priority	Each interrupt source is configured to have a priority (1 through 7). Smaller priority number has higher priority. In case of a tie, the interrupt source with smaller interrupt ID wins. Checker : Check if the resolution winner has highest priority amongst all the interrupts in the mie.				
1.4	Additonal Regs	1. Pending Reg (32'h80040084) 2. Claim/Complete Reg (32'h80040094)				
1.4.1	Pending Register	It indicate which registers are in the queue to be serviced. Checker: Check if PLIC clears the corresponding interrupt pending bit from mip reg				
1.4.2	Claim/Complete Reg	Core reads from this regsiter to claim an interrupt and accquire its source ID. Core writes to it to claim completion of a request. Checker: Check if ID of the interrupt claimed matches the interrupt ID predicted by the checker. Check if writedata (ID) matches the source ID present in the reg when completion is claimed. Check if the interrupt was requested and was claimed when the core claims its completion.				
2	Interrupt Priority Resolution	Testing the priority resolution when we have multiple interrupts				
2.1	Single Interrupt source					
2.2	Multiple Interrupts					
2.2.1	Distinct priorities	Each interrupt source has distinct priority				
2.2.2	Overlaps in priority	Some interrupt sources have overlapping/same priorities. Source with smaller source ID wins.				
2.3	Multiple Interrupts + Repeat	Another interrupt (same source) before current interrupt is serviced				
2.3.1	Before interrupt claimed	Regardless of whether the interrupt source is next in queue to be serviced				
2.3.2	After interrupt claim, but before service completion					
2.3.3	After service completion	ToDo: Need to check if interrupt inputs (GPIO) can/should be changed after 1 cycle (latch)				
3	Interrupt sources	Number of interrupt sources				
3.1	Multiple of 32					
3.2	Not multiple of 32					
4	AHB	TBD				
5	GPIO	TBD				
6	Change configuration midway through	Change config (enable, threshold) before or after interrupt claim, but before the core claims interrupt serivce completion Checker: TBD				
7	Claim complete source ID mismatch	Write value in service completion claim doesn't match the interrupt source ID currently being serviced.				
8	Not sure if these should be part of checks or tests	ToDo: Might need further discussion to decide upon the following, Interrupt request sent to the core. Core does not claim the request. Interrupt request claimed but never serviced by the core. Interrupt never requested, but service completion signaled (via dummy write to Claim reg). Interrupt requested and service completion signaled before interrupt is claimed.				