# Generic Interrupt Controller (INTC) Specification

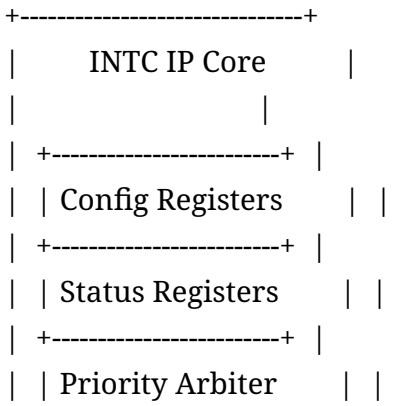
## 1. Overview

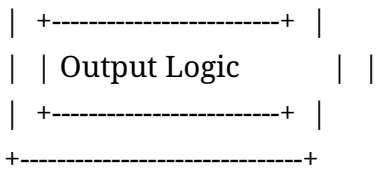
The Generic Interrupt Controller (INTC) is a highly configurable hardware IP core designed to manage multiple interrupt sources in a digital system. It provides a unified, programmable interface for interrupt detection, prioritization, masking, and output signaling. The INTC is suitable for a wide range of applications, from microcontroller subsystems to complex SoCs, and is optimized for flexibility, performance, and ease of integration.

## 2. Key Features

- Parameterized number of interrupt sources (N), configurable at synthesis time.
- Individual enable/disable control for each interrupt source via register.
- Mask feature: Each source can be masked via register, preventing it from propagating to the output interrupt.
- Configurable, programmable priority for each interrupt source.
- Status register indicating pending interrupts for each source.
- Configurable output interrupt polarity: active-high or active-low.
- Configurable output mode: level or pulse. In pulse mode, pulse width is programmable.
- Output interrupt and status can be cleared using a source-specific clear signal.

## 3. Block Diagram






## 4. Interface

The INTC IP core provides the following interface signals:

| Signal | Direction | Width | Description |
|---|---|---|---|
| clk | Input | 1 | System clock |
| rst_n | Input | 1 | Active-low reset |
| int_in[N-1:0] | Input | N | Interrupt source inputs |
| int_enable[N-1:0] | Input/Reg | N | Enable bits for each source |
| int_mask[N-1:0] | Input/Reg | N | Mask bits for each source |
| int_priority[N-1:0][P-1:0] | Input/Reg | N×P | Priority value for each source |
| int_clear[N-1:0] | Input/Reg | N | Clear signal for each source |
| out_mode | Input/Reg | 1 | Output mode select (0=level, 1=pulse) |
| out_polarity | Input/Reg | 1 | Output polarity select (0=active-low, 1=active-high) |
| pulse_width[W-1:0] | Input/Reg | W | Pulse width for output interrupt in pulse mode |

| | | | |
|---|---|---|---|
| int_status[N-1:0] | Output | N | Status register, 1=pending interrupt |
| int_out | Output | 1 | Output interrupt pin |
| int_vector[log2(N)-1:0] | Output | log2(N) | Index of highest-priority pending interrupt (optional) |

## 5. Register Map (Example)

| Register Name | Width | Access | Description |
|---|---|---|---|
| INT_ENABLE | N | RW | Enable bits for each source |
| INT_MASK | N | RW | Mask bits for each source |
| INT_PRIORITY | N×P | RW | Priority value for each source |
| INT_STATUS | N | RO | Status: 1=pending interrupt |
| INT_CLEAR | N | WO | Write 1 to clear corresponding status bit |
| OUT_MODE | 1 | RW | 0=level, 1=pulse |
| OUT_POLARITY | 1 | RW | 0=active-low, 1=active-high |
| PULSE_WIDTH | W | RW | Pulse width in clock cycles (pulse mode) |

## 6. Operation

- Each interrupt source is monitored. If enabled and not masked, a pending interrupt is set when the input is asserted.
- When multiple interrupts are pending, the source with the highest programmable priority is selected for output.
- Output generation: In level mode, output is asserted as long as any unmasked, enabled, pending interrupt exists. In pulse mode, output is asserted as a pulse of programmable width when a new interrupt is detected.
- Output polarity is configurable as active-high or active-low.
- Writing to the clear register for a source clears its pending status and, if no other interrupts are pending, deasserts the output.

## 7. Example Use Case

A system with 16 interrupt sources, each with enable, mask, and priority registers. Output interrupt is active-high, pulse mode, 4-cycle pulse width. Software can clear individual interrupts by writing to the INT_CLEAR register.

## 8. Notes

- All registers are accessible via a standard bus interface (e.g., APB, AXI-lite, or custom).
- All features are fully parameterizable for synthesis-time flexibility.
- The INTC is suitable for both simple and complex SoC designs.