#### Interrupt Collector Interface (InterruptCollectorIfc)

Interface containing a basic Interrupt Collector block.

|  |  |  |
| --- | --- | --- |
| Blocks of Interrupt Collector Interface | | |
| Block Address | ID | Block Name |
| 0x00 | InterruptCollectorBlk | [Interrupt Collector Block](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk) |

|  |  |
| --- | --- |
| Resets of Registers of Interrupt Collector Interface | |
| ID | Reset Name |
| Async | BusReset: Asynchronous Bus Reset |

##### Interrupt Collector Block (InterruptCollectorBlk)

This block defines a basic interrupt collector for \*\*level triggered\*\* interrupt sources.  
Usually edge triggered sources e.g., timer pulses can be converted to level triggered   
ones by catching them in the user logic.   
   
Constraints:  
   
1. Allow interrupt processing by multiple CPUs without need for spinlocks.  
2. Enable forwarding an interrupt to the CPU(s) by a mask for each source.  
3. Provide control pulses to notify the user logic when a interrupt service for an   
 interrupt request has been started and has been ended.   
4. Use the control pulses to reset the interrupt request fo a source or do it by   
 a write or read access directly to the user logic e.g. reading the receive data   
 register of an UART.   
   
Interrupt Collector details:  
   
.. figure:: resources/InterruptCollectorUserLogic.png  
 :scale: 50  
   
 Interrupt Collector details

|  |  |  |
| --- | --- | --- |
| Registers or Delegates of Interrupt Collector Block | | |
| 0x00 .. 0x0b | | [Interrupt Collector Block](#ifc:InterruptCollectorIfc/blk-lst:InterruptCollectorBlk) |
| Address | ID | Name |
| 0x00 | InterruptMaskReg | [Interrupt Mask Register](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk/fld:InterruptMaskReg) |
| 0x04 | InterruptRequestReg | [Interrupt Request Register](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk/fld:InterruptRequestReg) |
| 0x08 | InterruptServiceReg | [Interrupt Service Register](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk/fld:InterruptServiceReg) |

###### Interrupt Mask Register (InterruptMaskReg)

|  |  |  |  |
| --- | --- | --- | --- |
| Bits of Interrupt Mask Register | | | |
| 0x00 | | | [Interrupt Mask Register (InterruptMaskReg)](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk/fld-lst:InterruptMaskReg) |
| Bits | ID | Type | Description |
| 03 | Mask3 | RW | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | InterruptDisabled | RW | Interrupt is not forwarded to CPU(s). | | 0x1 | InterruptEnabled | RW | Interrupt is forwarded to CPU(s). |  |  |  |  |  | | --- | --- | --- | --- | | 0x0 | BusReset | RW | Default Bus Reset | |
| 02 | Mask2 | RW | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | InterruptDisabled | RW | Interrupt is not forwarded to CPU(s). | | 0x1 | InterruptEnabled | RW | Interrupt is forwarded to CPU(s). |  |  |  |  |  | | --- | --- | --- | --- | | 0x0 | BusReset | RW | Default Bus Reset | |
| 01 | Mask1 | RW | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | InterruptDisabled | RW | Interrupt is not forwarded to CPU(s). | | 0x1 | InterruptEnabled | RW | Interrupt is forwarded to CPU(s). |  |  |  |  |  | | --- | --- | --- | --- | | 0x0 | BusReset | RW | Default Bus Reset | |
| 00 | Mask0 | RW | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | InterruptDisabled | RW | Interrupt is not forwarded to CPU(s). | | 0x1 | InterruptEnabled | RW | Interrupt is forwarded to CPU(s). |  |  |  |  |  | | --- | --- | --- | --- | | 0x0 | BusReset | RW | Default Bus Reset | |

###### Interrupt Request Register (InterruptRequestReg)

|  |  |  |  |
| --- | --- | --- | --- |
| Bits of Interrupt Request Register | | | |
| 0x04 | | | [Interrupt Request Register (InterruptRequestReg)](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk/fld-lst:InterruptRequestReg) |
| Bits | ID | Type | Description |
| 03 | Request3 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | NotPending | R | An Interrupt is not pending. | | 0x1 | Pending | R | An Interrupt is pending. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been entered. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Request is set to NotPending after reset in the usually attached InterruptCollector HW block. | |
| 02 | Request2 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | NotPending | R | An Interrupt is not pending. | | 0x1 | Pending | R | An Interrupt is pending. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been entered. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Request is set to NotPending after reset in the usually attached InterruptCollector HW block. | |
| 01 | Request1 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | NotPending | R | An Interrupt is not pending. | | 0x1 | Pending | R | An Interrupt is pending. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been entered. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Request is set to NotPending after reset in the usually attached InterruptCollector HW block. | |
| 00 | Request0 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | NotPending | R | An Interrupt is not pending. | | 0x1 | Pending | R | An Interrupt is pending. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been entered. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Request is set to NotPending after reset in the usually attached InterruptCollector HW block. | |

###### Interrupt Service Register (InterruptServiceReg)

|  |  |  |  |
| --- | --- | --- | --- |
| Bits of Interrupt Service Register | | | |
| 0x08 | | | [Interrupt Service Register (InterruptServiceReg)](#ifc:InterruptCollectorIfc/blk:InterruptCollectorBlk/fld-lst:InterruptServiceReg) |
| Bits | ID | Type | Description |
| 03 | Service3 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | Ended | R | The Interrupt service has ended. | | 0x1 | InProcess | R | The Interrupt is in service. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been processed completely. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Service is set to Ended after reset in the usually attached InterruptCollector HW block. | |
| 02 | Service2 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | Ended | R | The Interrupt service has ended. | | 0x1 | InProcess | R | The Interrupt is in service. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been processed completely. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Service is set to Ended after reset in the usually attached InterruptCollector HW block. | |
| 01 | Service1 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | Ended | R | The Interrupt service has ended. | | 0x1 | InProcess | R | The Interrupt is in service. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been processed completely. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Service is set to Ended after reset in the usually attached InterruptCollector HW block. | |
| 00 | Service0 | R/W | |  |  |  |  | | --- | --- | --- | --- | | Value | ID | Type | Description | | 0x0 | Ended | R | The Interrupt service has ended. | | 0x1 | InProcess | R | The Interrupt is in service. | | 0x0 | NotConfirmed | W | Writing this value has no effect. | | 0x1 | Confirmed | W | Notifies HW that a pending interrupt has been recognized by SW.  SW confirms that a respective interrupt service routine has been processed completely. The value isn't stored, thus there is no need to reset it to '0' again.  Solely the write action is sufficient. |  |  |  |  |  | | --- | --- | --- | --- | | Init Value | ID | Impact | Description | | 0b0 | Async | R | BusReset: Asynchronous Bus Reset  The bit Service is set to Ended after reset in the usually attached InterruptCollector HW block. | |