Eccelerators Library IP specification

Table of Contents

List of Figures

List of Tables

1. EventCatcherIfc

1.1. Interrupt Collector Interface (InterruptCollectorIfc)

Interface containing a basic Interrupt Collector block.

Table 1.1. Blocks of Interrupt Collector Interface

Blocks of Interrupt Collector Interface		
Block Address	ID	Block Name
0x00	InterruptCollectorBlk	Interrupt Collector Block

Table 1.2. Resets of Interrupt Collector Interface

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

1.1.1. 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 1.1. Interrupt Collector details

Table 1.3. Registers or Delegates of Interrupt Collector Block

Registers or Delegates of Interrupt Collector Block		
0x00 .. 0x0b		Interrupt Collector Block
Address	ID	Name
0x00	InterruptMaskReg	Interrupt Mask Register
0x04	InterruptRequestReg	Interrupt Request Register
0x08	InterruptServiceReg	Interrupt Service Register

1.1.1.1. Interrupt Mask Register (InterruptMaskReg)

Table 1.4. Bits of Interrupt Mask Register

Bits of Interrupt Mask Register

0x00			Interrupt Mask Register (InterruptMaskReg)
Bits	ID	Type	Description
03	Mask3	RW	**Table 1.5. Values of Mask3** Value / ID / Type / Description 0x0 / InterruptDisabled / RW / Interrupt is not forwarded to CPU(s). 0x1 / InterruptEnabled / RW / Interrupt is forwarded to CPU(s). **Table 1.6. Resets of Mask3** 0x0 / BusReset / RW / Default Bus Reset
02	Mask2	RW	**Table 1.7. Values of Mask2** Value / ID / Type / Description 0x0 / InterruptDisabled / RW / Interrupt is not forwarded to CPU(s). 0x1 / InterruptEnabled / RW / Interrupt is forwarded to CPU(s). **Table 1.8. Resets of Mask2** 0x0 / BusReset / RW / Default Bus Reset
01	Mask1	RW	**Table 1.9. Values of Mask1** Value / ID / Type / Description 0x0 / InterruptDisabled / RW / Interrupt is not forwarded to CPU(s). 0x1 / InterruptEnabled / RW / Interrupt is forwarded to CPU(s). **Table 1.10. Resets of Mask1** 0x0 / BusReset / RW / Default Bus Reset
00	Mask0	RW	**Table 1.11. Values of Mask0** Value / ID / Type / Description 0x0 / InterruptDisabled / RW / Interrupt is not forwarded to CPU(s). 0x1 / InterruptEnabled / RW / Interrupt is forwarded to CPU(s). **Table 1.12. Resets of Mask0** 0x0 / BusReset / RW / Default Bus Reset

1.1.1.2. Interrupt Request Register (InterruptRequestReg)

Table 1.13. Bits of Interrupt Request Register

Bits of Interrupt Request Register			
0x04			Interrupt Request Register (InterruptRequestReg)
Bits	ID	Type	Description
03	Request3	R/W	**Table 1.14. Values of Request3** 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. **Table 1.15. Resets of Request3** 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	**Table 1.16. Values of Request2** 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.

Table 1.17. Resets of Request2

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.

Bits	ID	Type	Description
01	Request1	R/W	**Table 1.18. Values of Request1**

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.

Table 1.19. Resets of Request1

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.

Bits	ID	Type	Description
00	Request0	R/W	**Table 1.20. Values of Request0**

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.

Table 1.21. Resets of Request0

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.

1.1.1.3. Interrupt Service Register (InterruptServiceReg)

Table 1.22. Bits of Interrupt Service Register

Bits of Interrupt Service Register			
0x08			Interrupt Service Register (InterruptServiceReg)
Bits	ID	Type	Description
03	Service3	R/W	**Table 1.23. Values of Service3**

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.

Table 1.24. Resets of Service3

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.

Bits	ID	Type	Description
02	Service2	R/W	**Table 1.25. Values of Service2**

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.

Table 1.26. Resets of Service2

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	**Table 1.27. Values of Service1**

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.

Table 1.28. Resets of Service1

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	**Table 1.29. Values of Service0**

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.

Table 1.30. Resets of Service0

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.