

1.1 Interrupt Condition for Interconnect

Each CBASS crossbar IP may generate up to two interrupts:

- Default_err_intr: this interrupt is intent for the debug purpose. Any illegal transactions will be logged by CBASS crossbar IP and the illegal transaction information is logged by CBASS IP.
- Default_exp: this interrupt is asserted whenever there is a firewall violation. Only the CBASS IP contains firewall module can generate this interrupt.

The illegal transactions triggering default_err_intr can be caused by multiple reasons:

- The transaction is sent to a memory region not implemented in CBASS
- The target interface is in disabled state or the target peripheral is in disabled state

When CBASS detects an illegal transaction, it terminates the transaction gracefully to avoid system hang. In addition, the CBASS sends return status back to the initiator with error status. If this illegal transaction is a read transaction, the read data is retuned with all zero. If this illegal transaction is a write transaction, this write transaction is terminated and return write error back to initiator. The interrupt and illegal transaction logging can be found in the CBASS's err_regs region.

For any transaction is blocked by firewall, the transaction is also terminated gracefully and initiator will receive error status for the blocked transaction. If this blocked transaction is read, the read data is retuned with all zero. If this blocked transaction is a write transaction, this write transaction is terminated and return write error back to initiator. The default_exp interrupt and the blocked transaction information can be found in glb_regs region.

1.1.1 Register MMR for Default_err_intr

Each CBASS support default_err_intr contains err_regs region, which includes the following registers:

Address Offset	Register Mnemonic	Register Name
0x0	PID	Revision Register
0x4	DESTINATION_ID	Destination ID Register
0x24	EXCEPTION_LOGGING_HEADER0	Exception Logging Header 0 Register
0x28	EXCEPTION_LOGGING_HEADER1	Exception Logging Header 1 Register
0x2c	EXCEPTION_LOGGING_DATA0	Exception Logging Data 0 Register
0x30	EXCEPTION_LOGGING_DATA1	Exception Logging Data 1 Register

0x34	EXCEPTION_LOGGING_DATA2	Exception Logging Data 2 Register
0x38	EXCEPTION_LOGGING_DATA3	Exception Logging Data 3 Register
0x50	ERR_INTR_RAW_STAT	Global Interrupt Raw Status Register
0x54	ERR_INTR_ENABLED_STAT	Global Interrupt Enabled Status Register
0x58	ERR_INTR_ENABLE_SET	Interrupt Enable Set Register
0x5c	ERR_INTR_ENABLE_CLR	Interrupt Enable Clear Register
0x60	ERR_EOI	EOI Register

PID (Revision Register) is located at offset 0x0. The following shows the definition of each field of this register.

The Revision Register contains the major and minor revisions for the module.

Bit	Name	Type	Reset	Description
31:30	scheme	r	0x1	PID register scheme
29:28	bu	r	0x2	Business Unit: 10 = Processors
27:16	func	r	0x600	Module ID
15:11	rtl	r	0x6	RTL revision. Will vary depending on release.
10:8	major	r	0x1	Major revision
7:6	custom	r	0x0	Custom
5:0	minor	r	0x2	Minor revision

DESTINATION_ID (Destination ID Register) has offset 0x4.

The Destination ID Register defines the destination ID value for error messages.

Bit	Name	Type	Reset	Description
31:8	reserved	r	0x0	Reserved

7:0	dest_id	rw	0x0	The destination ID.
-----	---------	----	-----	---------------------

EXCEPTION_LOGGING_HEADER0 (Exception Logging Header 0 Register), (0x24)

The Exception Logging Header 0 Register contains the first word of the header.

Bit	Name	Type	Reset	Description
31:24	type_f	r	0x0	Type. 7 = CBASS.
23:8	src_id	r	0x0	Source ID. Always 0.
7:0	dest_id	r	0x0	Destination ID.

EXCEPTION_LOGGING_HEADER1 (Exception Logging Header 1 Register) has offset 0x28.

The Exception Logging Header 1 Register contains the second word of the header.

Bit	Name	Type	Reset	Description
31:24	group	r	0x0	Group. Always 0.
23:16	code	r	0x0	Code. 0 = CBASS decode error.
15:0	reserved	r	0x0	Reserved

EXCEPTION_LOGGING_DATA0 (Exception Logging Data 0 Register) has offset 0x2c.

The Exception Logging Data 0 Register contains the first word of the data.

Bit	Name	Type	Reset	Description
31:0	addr_l	r	0x0	Address lower 32 bits.

EXCEPTION_LOGGING_DATA1 (Exception Logging Data 1 Register), (0x30)

The Exception Logging Data 1 Register contains the second word of the data.

Bit	Name	Type	Reset	Description
31:16	reserved	r	0x0	Reserved
15:0	addr_h	r	0x0	Address upper 16 bits.

EXCEPTION_LOGGING_DATA2 (Exception Logging Data 2 Register has offset 0x34.

The Exception Logging Data 2 Register contains the third word of the data.

Bit	Name	Type	Reset	Description
31:28	reserved	r	0x0	Reserved
27:16	routeid	r	0x0	Route ID.
15:14	reserved	r	0x0	Reserved
13	write	r	0x0	Write.
12	read	r	0x0	Read.
11	debug	r	0x0	Debug.
10	cacheable	r	0x0	Cacheable.
9	priv	r	0x0	Priv.
8	secure	r	0x0	Secure.
7:0	priv_id	r	0x0	Priv ID.

EXCEPTION_LOGGING_DATA3 (Exception Logging Data 3 Register) has offset 0x38.

The Exception Logging Data 3 Register contains the fourth word of the data.

Bit	Name	Type	Reset	Description
31:10	reserved	r	0x0	Reserved
9:0	bytecnt	r	0x0	Byte count.

ERR_INTR_RAW_STAT (Global Interrupt Raw Status Register), (0x50)

The interrupt raw status register indicates if there is null interrupt regardless of interrupt enable

Bit	Name	Type	Reset	Description
31:1	reserved	r	0x0	Reserved
0	intr	rw1ts	0x0	Level Interrupt status

ERR_INTR_ENABLED_STAT (Global Interrupt Enabled Status Register), (0x54)

The interrupt status register is gated by the interrupt enable

Bit	Name	Type	Reset	Description
31:1	reserved	r	0x0	Reserved
0	enabled_intr	rw1tc	0x0	Level Enabled Interrupt status

ERR_INTR_ENABLE_SET (Interrupt Enable Set Register), (0x58)

Only when this register is set, null access will cause interrupt to be generated.

Bit	Name	Type	Reset	Description
31:1	reserved	r	0x0	Reserved
0	intr_enable_set	rw1ts	0x0	Interrupt Enable Set Register

ERR_INTR_ENABLE_CLR (Interrupt Enable Clear Register), (0x5c)

Setting this register disables the null interrupt generation

Bit	Name	Type	Reset	Description
31:1	reserved	r	0x0	Reserved
0	intr_enable_clr	rw1tc	0x0	Interrupt Enable Clear Register

ERR_EOI (EOI Register), (0x60)

Writing to EOI Register indicates that current interrupt has been serviced which then allows next interrupt to be generated

Bit	Name	Type	Reset	Description
-----	------	------	-------	-------------

31:16	reserved	r	0x0	Reserved
15:0	eoi_wr	rw	0x0	End Of Interrupt Register

1.1.2 Register MMR for Default_exp

All the register for default_exp_intr is located in the glb_regs region of the CBASS supports this feature. For each glb_reg region, the following registers are supported.

**Table 1:glb_regs for default_exp
_intr**

Address Offset	Register Mnemonic	Register Name
0x0	PID	Revision Register
0x4	DESTINATION_ID	Destination ID Register
0x20	EXCEPTION_LOGGING_CONTROL	Exception Logging Control Register
0x24	EXCEPTION_LOGGING_HEADER0	Exception Logging Header 0 Register
0x28	EXCEPTION_LOGGING_HEADER1	Exception Logging Header 1 Register
0x2c	EXCEPTION_LOGGING_DATA0	Exception Logging Data 0 Register
0x30	EXCEPTION_LOGGING_DATA1	Exception Logging Data 1 Register
0x34	EXCEPTION_LOGGING_DATA2	Exception Logging Data 2 Register
0x38	EXCEPTION_LOGGING_DATA3	Exception Logging Data 3 Register
0x40	EXCEPTION_PEND_SET	Exception Logging Pending Set Register
0x44	EXCEPTION_PEND_CLEAR	Exception Logging Pending Clear Register

The Revision Register contains the major and minor revisions for the module.

Bit	Name	Type	Reset	Description
31:30	scheme	r	0x1	PID register scheme
29:28	bu	r	0x2	Business Unit: 10 = Processors
27:16	func	r	0x600	Module ID
15:11	rtl	r	0x6	RTL revision. Will vary depending on release.
10:8	major	r	0x1	Major revision
7:6	custom	r	0x0	Custom
5:0	minor	r	0x2	Minor revision

The Destination ID Register defines the destination ID value for error messages.

Bit	Name	Type	Reset	Description
31:8	reserved	r	0x0	Reserved
7:0	dest_id	rw	0x0	The destination ID.

The Exception Logging Control Register controls the exception logging.

Bit	Name	Type	Reset	Description
31:2	reserved	r	0x0	Reserved
1	disable_pend	rw	0x0	Disables logging pending when set.
0	disable_f	rw	0x0	Disables logging when set.

The Exception Logging Header 0 Register contains the first word of the header.

Bit	Name	Type	Reset	Description
31:24	type_f	r	0x0	Type.
23:8	src_id	r	0x0	Source ID.
7:0	dest_id	r	0x0	Destination ID.

The Exception Logging Header 1 Register contains the second word of the header.

Bit	Name	Type	Reset	Description
31:24	group	r	0x0	Group.
23:16	code	r	0x0	Code.
15:0	reserved	r	0x0	Reserved

The Exception Logging Data 0 Register contains the first word of the data.

Bit	Name	Type	Reset	Description
31:0	addr_l	r	0x0	Address lower 32 bits.

The Exception Logging Data 1 Register contains the second word of the data.

Bit	Name	Type	Reset	Description
31:16	reserved	r	0x0	Reserved
15:0	addr_h	r	0x0	Address upper 16 bits.

The Exception Logging Data 2 Register contains the third word of the data.

Bit	Name	Type	Reset	Description
31:28	reserved	r	0x0	Reserved
27:16	routeid	r	0x0	Route ID.
15:14	reserved	r	0x0	Reserved
13	write	r	0x0	Write.
12	read	r	0x0	Read.
11	debug	r	0x0	Debug.
10	cacheable	r	0x0	Cacheable.
9	priv	r	0x0	Priv.
8	secure	r	0x0	Secure.
7:0	priv_id	r	0x0	Priv ID.

The Exception Logging Data 3 Register contains the fourth word of the data.

Bit	Name	Type	Reset	Description
31:10	reserved	r	0x0	Reserved
9:0	bytecnt	r	0x0	Byte count.

The Exception Logging Pending Set Register allows to set the pend signal.

Bit	Name	Type	Reset	Description
31:1	reserved	r	0x0	Reserved
0	pend_set	rw1ts	0x0	Write a 1 to set the exception pend signal.

The Exception Logging Pending Clear Register allows to clear the pend signal.

Bit	Name	Type	Reset	Description
31:1	reserved	r	0x0	Reserved
0	pend_clr	rw1tc	0x0	Write a 1 to clear the exception pend signal.