ERR<n>CTLR, Error Record <n> Control Register, n = 0 - 65534

The ERR<n>CTLR characteristics are:

Purpose

The error control register contains enable bits for the node that writes to this record:

- Enabling error detection and correction.
- Enabling the critical error, error recovery, and fault handling interrupts.
- Enabling in-band error response for uncorrected errors.

For each bit, if the node does not support the feature, then the bit is res0. The definition of each record is implementation defined.

Configuration

This register is present only when error record <n> is implemented and error record <n> is the first error record owned by a node. Otherwise, direct accesses to ERR<n>CTLR are res0.

ERR<n>FR describes the features implemented by the node.

Attributes

ERR<n>CTLR is a 64-bit register.

Field descriptions

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35
IMPLEMENTATION DEFINED																												
RES0																WDFI	Bit[14]	CI	CED	WDUI	Bit[10]	WCFI	Bit[8]	WUE	WFI	WUI	Bit[4]	Bit[3
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3

IMPLEMENTATION DEFINED, bits [63:32]

Reserved for implementation defined controls. Must permit SBZP write policy for software.

Bits [31:16]

Reserved, res0.

WDFI, bit [15]
When RAS System Architecture v2 is implemented and ERR<n>FR.DFI == 0b11:

Fault handling interrupt for Deferred errors on writes enable, with ERR<n>CTLR.WFI.

When enabled by ERR<n>CTLR.{WDFI, `WFI`}:

- The fault handling interrupt is generated for errors recorded as Deferred error on writes.
- If the corresponding fault handling interrupt control for corrected error events, ERR<n>CTLR.WCFI, is not implemented, then the fault handling interrupt is generated for corrected error events on writes.

WDFI	Meaning
`0b0`	When ERR<n>CTLR.WFI == 0, Fault handling interrupt not generated for Deferred errors on writes. When ERR<n>CTLR.WFI == 1, Fault handling interrupt generated for Deferred errors on writes.
`0b1`	When ERR<n>CTLR.WFI == 0, Fault handling interrupt generated for Deferred errors on writes. When ERR<n>CTLR.WFI == 1, Fault handling interrupt not generated for Deferred errors on writes.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bit[14]
When RAS System Architecture v2 is implemented and ERR<n>FR.DFI == 0b10:

DFI, bit [14]

Fault handling interrupt for Deferred errors enable, with ERR<n>CTLR.FI.

When ERR<n>FR.DFI == `0b10`, this control applies to errors on both reads and writes.

When enabled by ERR<n>CTLR.{DFI, FI}:

- The fault handling interrupt is generated for all errors recorded as Deferred error.
- If the fault handling interrupt control for corrected error events, ERR<n>CTLR.CFI, is not implemented, then the fault handling interrupt is generated for all corrected error events.

DFI	Meaning
`0b0`	When ERR<n>CTLR.FI == 0, Fault handling interrupt not generated for Deferred errors. When ERR<n>CTLR.FI == 1, Fault handling interrupt generated for Deferred errors.
`0b1`	When ERR<n>CTLR.FI == 0, Fault handling interrupt generated for Deferred errors. When ERR<n>CTLR.FI == 1, Fault handling interrupt not generated for Deferred errors.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

When RAS System Architecture v2 is implemented and ERR<n>FR.DFI == 0b11:

RDFI, bit [14]

Fault handling interrupt for Deferred errors on reads enable, with ERR<n>CTLR.RFI.

When ERR<n>FR.DFI == `0b11`, this field is named RDFI.

When enabled by ERR<n>CTLR.{RDFI, RFI}:

- The fault handling interrupt is generated for errors recorded as Deferred error on reads.
- If the corresponding fault handling interrupt control for corrected error events, ERR<n>CTLR.RCFI, is not implemented, then the fault handling interrupt is generated for corrected error events on reads.

RDFI	Meaning
`0b0`	When ERR<n>CTLR.RFI == 0, Fault handling interrupt not generated for Deferred errors on reads. When ERR<n>CTLR.RFI == 1, Fault handling interrupt generated for Deferred errors on reads.
`0b1`	When ERR<n>CTLR.RFI == 0, Fault handling interrupt generated for Deferred errors on reads. When ERR<n>CTLR.RFI == 1, Fault handling interrupt not generated for Deferred errors on reads.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

CI, bit [13]
When ERR<n>FR.CI == 0b10:

Critical error interrupt enable. When enabled, the critical error interrupt is generated for a critical error condition.

CI	Meaning
0b0	Critical error interrupt not generated for critical errors. Critical errors are treated as Uncontained errors.
0b1	Critical error interrupt generated for critical errors.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

CED, bit [12]
When RAS System Architecture v2 is implemented, ERR<n>FR.CEC != 0b000 and ERR<n>FR.CED == 1:

Disable generation of corrected error events from error counters.

CED	Meaning
0b0	Corrected error events are generated by the error counter or counters.
0b1	Corrected error events are generated when a Corrected error is recorded.

See ERR<n>CTLR.CFI for more information on corrected error events.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

WDUI, bit [11]
When ERR<n>FR.DUI == 0b11:

Error recovery interrupt for Deferred errors on writes enable.

When enabled, the error recovery interrupt is generated for errors recorded as Deferred error on writes.

WDUI	Meaning
0b0	Error recovery interrupt not generated for Deferred errors on writes.
0b1	Error recovery interrupt generated for Deferred errors on writes.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bit[10]
When ERR<n>FR.DUI == 0b10:

DUI, bit [10]

Error recovery interrupt for Deferred errors enable.

When ERR<n>FR.DUI == 0b10, this control applies to errors arising from both reads and writes.

When enabled, the error recovery interrupt is generated for all errors recorded as Deferred error.

DUI	Meaning
0b0	Error recovery interrupt not generated for Deferred errors.
0b1	Error recovery interrupt generated for Deferred errors.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

When ERR<n>FR.DUI == 0b11:

RDUI, bit [10]

Error recovery interrupt for Deferred errors on reads enable.

When ERR<n>FR.DUI == `0b11`, this field is named RDUI.

When enabled, the error recovery interrupt is generated for errors recorded as Deferred error on reads.

RDUI	Meaning
`0b0`	Error recovery interrupt not generated for Deferred errors on reads.
`0b1`	Error recovery interrupt generated for Deferred errors on reads.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

WCFI, bit [9]
When ERR<n>FR.CFI == 0b11:

Fault handling interrupt for corrected error events on writes enable.

When enabled, the fault handling interrupt is generated for corrected error events on writes.

WCFI	Meaning

0b0	Fault handling interrupt not generated for corrected error events on writes.
0b1	Fault handling interrupt generated for corrected error events on writes.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bit[8]
When ERR<n>FR.CFI == 0b10:

CFI, bit [8]

Fault handling interrupt for corrected error events enable.

When ERR<n>FR.CFI == 0b10, this control applies to errors on both reads and writes.

When enabled, the fault handling interrupt is generated for all corrected error events.

CFI	Meaning
0b0	Fault handling interrupt not generated for corrected error events.
0b1	Fault handling interrupt generated for corrected error events.

If the node implements a corrected error counter or counters, and either ERR<n>CTLR.CED is not implemented or ERR<n>CTLR.CED is 0, then a corrected error event is defined as follows:

- A corrected error event occurs when a counter overflows and sets a counter overflow flag to 1.
- It is unpredictable whether a corrected error event occurs when a software write sets a counter overflow flag to 1.

- It is unpredictable whether a corrected error event occurs when a counter overflows and the overflow flag was previously set to 1.

Otherwise, a corrected error event occurs when the error record records an error as a Corrected error.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

When ERR<n>FR.CFI == 0b11:

RCFI, bit [8]

Fault handling interrupt for corrected error events on reads enable.

When ERR<n>FR.CFI == `0b11`, this field is named RCFI.

When enabled, the fault handling interrupt is generated for corrected error events on reads.

RCFI	Meaning
`0b0`	Fault handling interrupt not generated for corrected error events on reads.
`0b1`	Fault handling interrupt generated for corrected error events on reads.

If the node implements a corrected error counter or counters, and either ERR<n>CTLR.CED is not implemented or ERR<n>CTLR.CED is 0, then a corrected error event is defined as follows:

- A corrected error event occurs when a counter overflows and sets a counter overflow flag to 1.
- It is unpredictable whether a corrected error event occurs when a software write sets a counter overflow flag to 1.
- It is unpredictable whether a corrected error event occurs when a counter overflows and the overflow flag was previously set to 1.

Otherwise, a corrected error event occurs when the error record records an error as a Corrected error.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

WUE, bit [7]
When ERR<n>FR.UE == 0b11:

In-band error response on writes enable.

When enabled, responses to writes that detect an error that is not corrected and is not deferred are signaled with an in-band error response (External Abort).

It is implementation defined whether an uncorrected error that is deferred and recorded as Deferred error, but is not deferred to the Requester, will signal an in-band error response to the Requester.

WUE	Meaning
0b0	In-band error response for uncorrected errors on writes disabled.
0b1	In-band error response for uncorrected errors on writes enabled.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

WFI, bit [6]
When ERR<n>FR.FI == 0b11:

Fault handling interrupt on writes enable.

When enabled:

- The fault handling interrupt is generated for errors recorded as Uncorrected error on writes.
- If the corresponding fault handling interrupt control for Deferred errors, ERR<n>CTLR.WDFI, is not implemented, then the fault handling interrupt is generated for errors recorded as Deferred error on writes.
- If the corresponding fault handling interrupt controls for Deferred errors and corrected error events, ERR<n>CTLR.{WDFI, WCFI}, are not implemented, then the fault handling interrupt is generated for corrected error events on writes.

WFI	Meaning
0b0	Fault handling interrupt on writes disabled.
0b1	Fault handling interrupt on writes enabled.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

WUI, bit [5]
When ERR<n>FR.UI == 0b11:

Uncorrected error recovery interrupt on writes enable.

When enabled, the error recovery interrupt is generated for errors recorded as Uncorrected error on writes.

WUI	Meaning
0b0	Error recovery interrupt on writes disabled.
0b1	Error recovery interrupt on writes enabled.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bit[4]
When ERR<n>FR.UE == 0b10:

UE, bit [4]

In-band error response enable.

When ERR<n>FR.UE == `0b10`, this control applies to errors arising from both reads and writes.

When enabled, responses to transactions that detect an error that is not corrected and is not deferred are signaled with an in-band error response (External Abort).

It is implementation defined whether an uncorrected error that is deferred and recorded as Deferred error, but is not deferred to the Requester, will signal an in-band error response to the Requester.

UE	Meaning
`0b0`	In-band error response for uncorrected errors disabled.
`0b1`	In-band error response for uncorrected errors enabled.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

When ERR<n>FR.UE == 0b11:

RUE, bit [4]

In-band error response on reads enable.

When ERR<n>FR.UE == `0b11`, this field is named RUE.

When enabled, responses to reads that detect an error that is not corrected and is not deferred are signaled with an in-band error response (External Abort).

It is implementation defined whether an uncorrected error that is deferred and recorded as Deferred error, but is not deferred to the Requester, will signal an in-band error response to the Requester.

RUE	Meaning
0b0	In-band error response for uncorrected errors on reads disabled.
0b1	In-band error response for uncorrected errors on reads enabled.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bit[3]
When ERR<n>FR.FI == 0b10:

FI, bit [3]

Fault handling interrupt enable.

When ERR<n>FR.FI == 0b10, this control applies to errors on both reads and writes.

When enabled:

- The fault handling interrupt is generated for all errors recorded as Uncorrected error.
- If the fault handling interrupt control for Deferred errors, ERR<n>CTLR.DFI, is not implemented, then the fault handling interrupt is generated for all errors recorded as Deferred error.
- If the fault handling interrupt controls for Deferred errors and corrected error events, ERR<n>CTLR.{DFI, CFI}, are not implemented, then the fault handling interrupt is generated for all corrected error events.

FI	Meaning
0b0	Fault handling interrupt disabled.
0b1	Fault handling interrupt enabled.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

When ERR<n>FR.FI == 0b11:

RFI, bit [3]

Fault handling interrupt on reads enable.

When ERR<n>FR.FI == `0b11`, this field is named RFI.

When enabled:

- The fault handling interrupt is generated for errors recorded as Uncorrected error on reads.
- If the corresponding fault handling interrupt control for Deferred errors, ERR<n>CTLR.RDFI, is not implemented, then the fault handling interrupt is generated for errors recorded as Deferred error on reads.
- If the corresponding fault handling interrupt controls for Deferred errors and corrected error events, ERR<n>CTLR.{RDFI, RCFI}, are not implemented, then the fault handling interrupt is generated for corrected error events on reads.

RFI	Meaning
`0b0`	Fault handling interrupt on reads disabled.
`0b1`	Fault handling interrupt on reads enabled.

See ERR<n>CTLR.CFI for more information on corrected error events.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bit[2]
When ERR<n>FR.UI == 0b10:

UI, bit [2]

Uncorrected error recovery interrupt enable.

When ERR<n>FR.UI == `0b10`, this control applies to errors arising from both reads and writes.

When enabled, the error recovery interrupt is generated for all errors recorded as Uncorrected error.

UI	Meaning
`0b0`	Error recovery interrupt disabled.
`0b1`	Error recovery interrupt enabled.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

When ERR<n>FR.UI == 0b11:

RUI, bit [2]

Uncorrected error recovery interrupt on reads enable.

When ERR<n>FR.UI == `0b11`, this field is named RUI.

When enabled, the error recovery interrupt is generated for errors recorded as Uncorrected error on reads.

RUI	Meaning
`0b0`	Error recovery interrupt on reads disabled.
`0b1`	Error recovery interrupt on reads enabled.

The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

IMPLEMENTATION DEFINED, bit [1]

Reserved for implementation defined controls. Must permit SBZP write policy for software.

ED, bit [0]
When ERR<n>FR.ED == 0b10:

Error reporting and logging enable. When disabled, the node behaves as if error detection and correction are disabled, and no errors are recorded or signaled by the node. Arm recommends that, when disabled, correct error detection and correction codes are written for writes, unless disabled by an implementation defined control for error injection.

ED	Meaning
0b0	Error reporting disabled.
0b1	Error reporting enabled.

It is implementation defined whether the node fully disables error detection and correction when reporting is disabled. That is, even with error reporting disabled, the node might continue to silently correct errors. Uncorrected errors might result in corrupt data being silently propagated by the node.

Note

If this node requires initialization after Cold reset to prevent signaling false errors, then Arm recommends this field is set to 0 on Cold reset, meaning errors are not reported from Cold reset. This allows boot software to initialize a node without signaling errors. Software can enable error reporting after the node is initialized. Otherwise, the Cold reset value is implementation defined. If the Cold reset value is 1, the reset values of other controls in this register are also implementation defined and should not be unknown.

The reset behavior of this field is:

- On an Error recovery reset, when RAS System Architecture v2 is implemented and ERR<n>FR.SRV == 1, this field resets to `0`.
- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<n>FR.SRV == 1, this field resets to `0`.
  - Otherwise, this field resets to an implementation defined value.

Otherwise:

Reserved, res0.

Accessing ERR<n>CTLR

ERR<n>CTLR can be accessed through the memory-mapped interfaces:

Component	Offset	Instance
RAS	`0x008` + (64 * n)	ERR<n>CTLR

Accesses on this interface are **RW**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94