ERR<n>FR, Error Record <n> Feature Register, n = 0 - 65534

The ERR<n>FR characteristics are:

Purpose

Defines whether error record <n> is the first record owned by a node:

- If error record <n> is the first error record owned by a node, then ERR<n>FR.ED is not `0b00`.
- If error record <n> is not the first error record owned by a node, then ERR<n>FR.ED is `0b00`.

If error record <n> is the first record owned by the node, defines which of the common architecturally-defined features are implemented by the node and, of the implemented features, which are software programmable.

Configuration

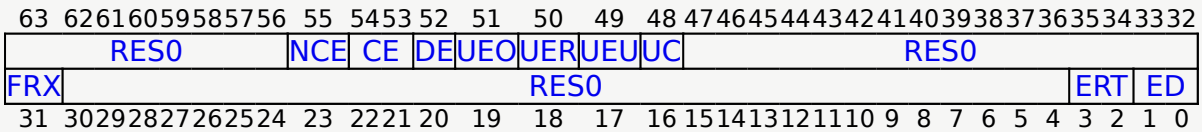
This register is present only when error record <n> is implemented. Otherwise, direct accesses to ERR<n>FR are res0.

Attributes

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

Field descriptions

When error record <n> is not implemented or error record <n> is not the first error record owned by the node:



Bits [63:56]

Reserved, res0.

NCE, bit [55]
When RAS System Architecture v2 is implemented and ERR<q>FR.CEC != 0b000:

No countable errors. Describes whether this error record supports recording countable errors. Defined values are:

NCE	Meaning
0b0	Records countable errors.
0b1	Does not record countable errors.

When ERR<q>FR.CEC != 0b000, at least one error record owned by the node records countable errors.

Otherwise:

Reserved, res0.

CE, bits [54:53]
When ERR<n>FR.FRX == 1:

Corrected Error recording. Describes the types of Corrected errors the error record can record, if any. Defined values are:

CE	Meaning
0b00	Does not record Corrected errors.
0b01	Records only transient or persistent Corrected errors. That is, Corrected errors recorded by setting ERR<n>STATUS.CE to either 0b01 or 0b11.
0b10	Records only non-specific Corrected errors. That is, Corrected errors recorded by setting ERR<n>STATUS.CE to 0b10.
0b11	Records all types of Corrected error.

Otherwise:

Reserved, res0.

DE, bit [52]
When ERR<n>FR.FRX == 1:

Deferred Error recording. Describes whether the error record supports recording Deferred errors. Defined values are:

DE	Meaning
0b0	Does not record Deferred errors.
0b1	Records Deferred errors.

Otherwise:

Reserved, res0.

UEO, bit [51]
When ERR<n>FR.FRX == 1:

Latent or Restartable Error recording. Describes whether the error record supports recording Latent or Restartable errors. Defined values are:

UEO	Meaning
0b0	Does not record Latent or Restartable errors.
0b1	Records Latent or Restartable errors.

Otherwise:

Reserved, res0.

UER, bit [50]
When ERR<n>FR.FRX == 1:

Signaled or Recoverable Error recording. Describes whether the error record supports recording Signaled or Recoverable errors. Defined values are:

UER	Meaning
0b0	Does not record Signaled or Recoverable errors.
0b1	Records Signaled or Recoverable errors.

Otherwise:

Reserved, res0.

UEU, bit [49]
When ERR<n>FR.FRX == 1:

Unrecoverable Error recording. Describes whether the error record supports recording Unrecoverable errors. Defined values are:

UEU	Meaning
0b0	Does not record Unrecoverable errors.
0b1	Records Unrecoverable errors.

Otherwise:

Reserved, res0.

UC, bit [48]
When ERR<n>FR.FRX == 1:

Uncontainable Error recording. Describes whether the error record supports recording Uncontainable errors. Defined values are:

UC	Meaning
0b0	Does not record Uncontainable errors.
0b1	Records Uncontainable errors.

Otherwise:

Reserved, res0.

Bits [47:32]

Reserved, res0.

FRX, bit [31]
When Error record <n> is implemented, RAS System Architecture v2 is implemented and ERR<n>FR.ERT == 0b00:

Feature Register extension. Defines whether ERR<n>FR[54:48] describe the error types supported by this error record. Defined values are:

FRX	Meaning
0b0	ERR<n>FR[54:48] are res0.
0b1	ERR<n>FR[54:48] are defined by the architecture.

If ERR<n>FR.FRX is 0, and error record <n> is implemented, then the error types supported by this error record are as described by the first error record of this node.

Otherwise:

Reserved, res0.

Bits [30:4]

Reserved, res0.

ERT, bits [3:2]
When RAS System Architecture v2 is implemented:

Error Record Type. Defines the type of error record. Defined values are:

ERT	Meaning
0b00	Error record <n> not implemented or is a normal record that is not the first error record of the node.
0b01	Error record <n> is a continuation record of the previous error record, <n-1>.

All other values are reserved.

Otherwise:

Reserved, res0.

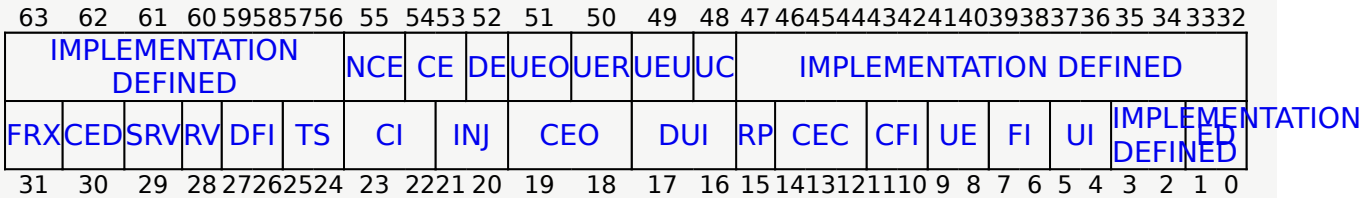
ED, bits [1:0]

Error reporting and logging. Indicates error record <n> is not the first error record owned the node.

ED	Meaning
0b00	Error record <n> is not implemented or is not the first error record owned by the node.

Access to this field is **RO**.

When error record <n> is the first error record owned by the node:



IMPLEMENTATION DEFINED, bits [63:56]
When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

NCE, bit [55]
When RAS System Architecture v2 is implemented and ERR<n>FR.CEC != 0b000:

NCE, bit [55:55]

No countable errors. Describes whether this error record supports recording countable errors. Defined values are:

NCE	Meaning
0b0	Records countable errors.
0b1	Does not record countable errors.

When ERR<n>FR.CEC != 0b000, at least one error record owned by the node records countable errors.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

IMPLEMENTATION DEFINED, bit [55:55]

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

CE, bits [54:53]
When ERR<n>FR.FRX == 1:

Corrected Error recording. Describes the types of Corrected errors the node can record, if any. Defined values are:

CE	Meaning
0b00	Does not record Corrected errors.

0b01	Records only transient or persistent Corrected errors. That is, Corrected errors recorded by setting ERR<n>STATUS.CE to either 0b01 or 0b11.
0b10	Records only non-specific Corrected errors. That is, Corrected errors recorded by setting ERR<n>STATUS.CE to 0b10.
0b11	Records all types of Corrected error.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

DE, bit [52]
When ERR<n>FR.FRX == 1:

Deferred Error recording. Describes whether the node supports recording Deferred errors. Defined values are:

DE	Meaning
0b0	Does not record Deferred errors.
0b1	Records Deferred errors.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

UEO, bit [51]
When ERR<n>FR.FRX == 1:

Latent or Restartable Error recording. Describes whether the node supports recording Latent or Restartable errors. Defined values are:

UEO	Meaning
0b0	Does not record Latent or Restartable errors.
0b1	Records Latent or Restartable errors.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

UER, bit [50]
When ERR<n>FR.FRX == 1:

Signaled or Recoverable Error recording. Describes whether the node supports recording Signaled or Recoverable errors. Defined values are:

UER	Meaning
0b0	Does not record Signaled or Recoverable errors.
0b1	Records Signaled or Recoverable errors.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

UEU, bit [49]
When ERR<n>FR.FRX == 1:

Unrecoverable Error recording. Describes whether the node supports recording Unrecoverable errors. Defined values are:

UEU	Meaning
0b0	Does not record Unrecoverable errors.

0b1	Records Unrecoverable errors.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

UC, bit [48]
When ERR<n>FR.FRX == 1:

Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors. Defined values are:

UC	Meaning
0b0	Does not record Uncontainable errors.
0b1	Records Uncontainable errors.

When RAS System Architecture v2 is not implemented and ERR<n>FR.FRX == 0:

Reserved for identifying implementation defined controls.

Otherwise:

Reserved, res0.

IMPLEMENTATION DEFINED, bits [47:32]

Reserved for identifying implementation defined controls.

FRX, bit [31]
When RAS System Architecture v1.1 is implemented:

Feature Register extension. Defines whether ERR<n>FR[63:48] are architecturally defined. Defined values are:

FRX	Meaning
0b0	ERR<n>FR[63:48] are implementation defined.
0b1	ERR<n>FR[63:48] are defined by the architecture.

Otherwise:

Reserved, res0.

CED, bit [30]
When RAS System Architecture v2 is implemented and ERR<n>FR.CEC != 0b000:

Error counter disable. Indicates whether the node implements a control to disable any implemented Corrected error counters. Defined values are:

CED	Meaning
0b0	Error counter disable control is not implemented and the error counter(s) are always enabled. ERR<n>CTLR.CED is res0.
0b1	Enabling and disabling of error counter(s) is supported and controlled by ERR<n>CTLR.CED.

Otherwise:

Reserved, res0.

SRV, bit [29]
When RAS System Architecture v2 is implemented:

Status Reset Value. Indicates how node <n> and each error record <m> owned by node <n> is reset. Defined values are:

SRV	Meaning
0b0	Node <n> and each error record <m> owned by node <n> are reset as follows: • ERR<m>STATUS.{AV, V, MV} are set to {0, 0, 0} on a Cold reset and preserved on Error Recovery reset. • ERR<n>CTLR.ED is set to an implementation defined value on a Cold reset and preserved on Error Recovery reset.

0b1	Node <n> and each error record <m> owned by node <n> are reset as follows: • ERR<m>STATUS.{AV, V, MV} are set to architecturally unknown values on a Cold reset and preserved on Error Recovery reset. • ERR<n>CTLR.ED is set to 0 on both Cold reset and Error Recovery reset.

All other values are reserved.

Otherwise:

Reserved, res0.

RV, bit [28]
When RAS System Architecture v2 is implemented:

Reset Valid. Indicates whether each error record <m> implemented by the node includes the Reset Valid flag, ERR<m>STATUS.RV. Defined values are:

RV	Meaning
0b0	ERR<m>STATUS.RV is res0.
0b1	ERR<m>STATUS.RV is a R/W1C bit set to 1 on Error Recovery reset.

All other values are reserved.

Otherwise:

Reserved, res0.

DFI, bits [27:26]
When RAS System Architecture v2 is implemented and ERR<n>FR.FI != 0b00:

Fault handling interrupt for deferred errors control. Indicates whether the enabling and disabling of fault handling interrupts on deferred errors is supported by the node. Defined values are:

DFI	Meaning

0b00	Does not support the enabling and disabling of fault handling interrupts on deferred errors. ERR<n>CTLR.DFI is res0.
0b10	Enabling and disabling of fault handling interrupts on deferred errors is supported and controllable using ERR<n>CTLR.DFI.
0b11	Enabling and disabling of fault handling interrupts on deferred errors is supported, and controllable using ERR<n>CTLR.WDFI for writes and ERR<n>CTLR.RDFI for reads.

All other values are reserved.

Otherwise:

Reserved, res0.

TS, bits [25:24]

Timestamp Extension. Indicates whether, for each error record <m> owned by this node, ERR<m>MISC3 is used as the timestamp register, and, if it is, the timebase used by the timestamp.

TS	Meaning
0b00	Does not support a timestamp register.
0b01	Implements a timestamp register in ERR<n>MISC3 for each error record <m> owned by the node. The timestamp uses the same timebase as the system Generic Timer. **Note** For an error record that has an affinity to a PE, this is the same timer that is visible through CNTPCT_EL0 at the highest Exception level on that PE.

0b10	Implements a timestamp register in ERR<m>MISC3 for each error record <m> owned by the node. The timestamp uses an implementation defined timebase.

All other values are reserved.

CI, bits [23:22]

Critical error interrupt. Indicates whether the critical error interrupt and associated controls are implemented by the node.

CI	Meaning
0b00	Does not support the critical error interrupt. ERR<n>CTLR.CI is res0.
0b01	Critical error interrupt is supported and always enabled. ERR<n>CTLR.CI is res0.
0b10	Critical error interrupt is supported and controllable using ERR<n>CTLR.CI.

All other values are reserved.

INJ, bits [21:20]

Fault Injection Extension. Indicates whether the Common Fault Injection Model Extension is implemented by the node.

INJ	Meaning
0b00	Does not support the Common Fault Injection Model Extension.
0b01	Supports the Common Fault Injection Model Extension. See ERR<n>PFGF for more information.

All other values are reserved.

CEO, bits [19:18]
When ERR<n>FR.CEC != 0b000:

Corrected Error overwrite. Indicates the behavior of the node when a second or subsequent Corrected error is recorded and a first Corrected error has previously been recorded by an error record <m> owned by the node.

CEO	Meaning

0b00	Keeps the previous error syndrome.
0b01	If ERR<m>STATUS.OF is 1 before the Corrected error is counted, then the error record keeps the previous syndrome. Otherwise the previous syndrome is overwritten.

All other values are reserved.

The second or subsequent Corrected error is counted by the Corrected error counter, regardless of the value of this field. If counting the error causes unsigned overflow of the counter, then ERR<m>STATUS.OF is set to 1.

This means that, if no other error is subsequently recorded that overwrites the syndrome:

- If ERR<n>FR.CEO is 0b00, the error record holds the syndrome for the first recorded Corrected error.
- If ERR<n>FR.CEO is 0b01, the error record holds the syndrome for the most recently recorded Corrected error before the counter overflows.

Otherwise:

Reserved, res0.

DUI, bits [17:16]
When ERR<n>FR.UI != 0b00:

Error recovery interrupt for deferred errors control. Indicates whether the enabling and disabling of error recovery interrupts on deferred errors is supported by the node.

DUI	Meaning
0b00	Does not support the enabling and disabling of error recovery interrupts on deferred errors. ERR<n>CTLR.DUI is res0.
0b10	Enabling and disabling of error recovery interrupts on deferred errors is supported and controllable using ERR<n>CTLR.DUI.

0b11	Enabling and disabling of error recovery interrupts on deferred errors is supported, and controllable using ERR<n>CTLR.WDUI for writes and ERR<n>CTLR.RDUI for reads.

All other values are reserved.

Otherwise:

Reserved, res0.

RP, bit [15]
When ERR<n>FR.CEC != 0b000:

Repeat counter. Indicates whether the node implements a second Corrected error counter in ERR<m>MISC0 for each error record <m> owned by the node that can record countable errors.

RP	Meaning
0b0	Implements a single Corrected error counter in ERR<m>MISC0 for each error record <m> owned by the node that can record countable errors.
0b1	Implements a first (repeat) counter and a second (other) counter in ERR<m>MISC0 for each error record <m> owned by the node that can record countable errors. The repeat counter is the same size as the primary error counter.

Otherwise:

Reserved, res0.

CEC, bits [14:12]

Corrected Error Counter. Indicates whether the node implements the standard format Corrected error counter mechanisms in ERR<m>MISC0 for each error record <m> owned by the node that can record countable errors.

CEC	Meaning

0b000	Does not implement the standard format Corrected error counter model.
0b010	Implements an 8-bit Corrected error counter in ERR<m>MISC0[39:32] for each error record <m> owned by the node that can record countable errors.
0b100	Implements a 16-bit Corrected error counter in ERR<m>MISC0[47:32] for each error record <m> owned by the node that can record countable errors.

All other values are reserved.

Note

Implementations might include other error counter models, or might include the standard format model and not indicate this in ERR<n>FR.

CFI, bits [11:10]
When ERR<n>FR.FI != 0b00:

Fault handling interrupt for corrected errors control. Indicates whether the enabling and disabling of fault handling interrupts on corrected errors is supported by the node.

CFI	Meaning
0b00	Does not support the enabling and disabling of fault handling interrupts on corrected errors. ERR<n>CTLR.CFI is res0.
0b10	Enabling and disabling of fault handling interrupts on corrected errors is supported and controllable using ERR<n>CTLR.CFI.
0b11	Enabling and disabling of fault handling interrupts on corrected errors is supported, and controllable using ERR<n>CTLR.WCFI for writes and ERR<n>CTLR.RCFI for reads.

All other values are reserved.

Otherwise:

Reserved, res0.

UE, bits [9:8]

In-band error response (External Abort). Indicates whether the in-band error response and associated controls are implemented by the node.

UE	Meaning
0b00	Does not support the in-band error response. ERR<n>CTLR.UE is res0.
0b01	In-band error response is supported and always enabled. ERR<n>CTLR.UE is res0.
0b10	In-band error response is supported and controllable using ERR<n>CTLR.UE.
0b11	In-band error response is supported, and controllable using ERR<n>CTLR.WUE for writes and ERR<n>CTLR.RUE for reads.

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.

FI, bits [7:6]

Fault handling interrupt. Indicates whether the fault handling interrupt and associated controls are implemented by the node.

FI	Meaning
0b00	Does not support the fault handling interrupt. ERR<n>CTLR.FI is res0.
0b01	Fault handling interrupt is supported and always enabled. ERR<n>CTLR.FI is res0.
0b10	Fault handling interrupt is supported and controllable using ERR<n>CTLR.FI.
0b11	Fault handling interrupt is supported, and controllable using ERR<n>CTLR.WFI for writes and ERR<n>CTLR.RFI for reads.

UI, bits [5:4]

Error recovery interrupt for uncorrected errors. Indicates whether the error handling interrupt and associated controls are implemented by the node.

UI	Meaning
0b00	Does not support the error handling interrupt. ERR<n>CTLR.UI is res0.
0b01	Error handling interrupt is supported and always enabled. ERR<n>CTLR.UI is res0.
0b10	Error handling interrupt is supported and controllable using ERR<n>CTLR.UI.
0b11	Error handling interrupt is supported, and controllable using ERR<n>CTLR.WUI for writes and ERR<n>CTLR.RUI for reads.

IMPLEMENTATION DEFINED, bits [3:2]

implementation defined.

ED, bits [1:0]

Error reporting and logging. Indicates error record <n> is a normal record and the first record owned the node, and whether the node implements the controls for enabling and disabling error reporting and logging. Defined values are:

ED	Meaning
0b01	Error reporting and logging always enabled. ERR<n>CTLR.ED is res0.
0b10	Error reporting and logging is controllable using ERR<n>CTLR.ED.

All other values are reserved.

When RAS System Architecture v2 is implemented and error record <n> is a proxy for a RAS agent:

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 34 33 32			
RES0			
RES0		ERT	ED
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 2 1 0			

Bits [63:4]

Reserved, res0.

ERT, bits [3:2]

Error Record Type. Defines the type of error record.

ERT	Meaning
0b01	Error record is a proxy for a RAS agent.

All other values are reserved.

Access to this field is **RO**.

ED, bits [1:0]

Error reporting and logging. Indicates error record <n> is not a true error record.

ED	Meaning
0b11	Error record <n> is not an error record.

Access to this field is **RO**.

Accessing ERR<n>FR

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

Component	Offset	Instance
RAS	0x000 + (64 * n)	ERR<n>FR

Accesses on this interface are **RO**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94