ERR<n>STATUS, Error Record <n> Primary Status Register, n = 0 - 65534

The ERR<n>STATUS characteristics are:

Purpose

When RAS System Architecture v2 is implemented, error record <n> might be one of the following:

- A continuation record containing more information about the error recorded in error record <n-1>. In this case, ERR<n>STATUS contains a subset of the values of a normal error record status register.
- A proxy for a different RAS agent. In this case, ERR<n>STATUS reports the status of the RAS agent.

Otherwise, ERR<n>STATUS contains status information for error record <n>, including:

- Whether any error has been detected (valid).
- Whether any detected error was not corrected, and returned to a Requester.
- Whether any detected error was not corrected and deferred.
- Whether an error record has been discarded because additional errors have been detected before the first error was handled by software (overflow).
- Whether any error has been reported.
- Whether the other error record registers contain valid information.
- Whether the error was reported because poison data was detected or because a corrupt value was detected by an error detection code.
- A primary error code.
- An implementation defined extended error code.

Within this register:

- ERR<n>STATUS.{AV, V, MV} are valid bits that define whether error record <n> registers are valid.
- ERR<n>STATUS.{UE, OF, CE, DE, UET} encode the types of error or errors recorded.
- ERR<n>STATUS.{CI, ER, PN, IERR, SERR} are syndrome fields.

Configuration

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

ERR<q>FR describes the features implemented by the node that owns error record <n>. <q> is the index of the first error record owned by the same node as error record <n>. If the node owns a single record then q = n.

For implementation defined fields in ERR<n>STATUS, writing zero returns the error record to an initial quiescent state.

In particular, if any implementation defined syndrome fields might generate a Fault Handling or Error Recovery Interrupt request, writing zero is sufficient to deactivate the Interrupt request.

Fields that are read-only, nonzero, and ignore writes are compliant with this requirement.

Note

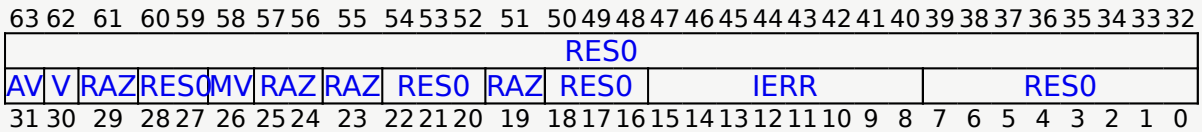
Arm recommends that any implementation defined syndrome field that can generate a Fault Handling, Error Recovery, Critical, or implementation defined, interrupt request is disabled at Cold reset and is enabled by software writing an implementation defined nonzero value to an implementation defined field in ERR<q>CTLR.

Attributes

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

Field descriptions

When RAS System Architecture v2 is implemented, ERR<q>FR.ED == 0b00 and ERR<q>FR.ERT == 0b01:



Bits [63:32]

Reserved, res0.

AV, bit [31]
When error record <n> includes an additional address associated with an error:

Address Valid.

AV	Meaning

0b0	ERR<n>ADDR not valid.
0b1	ERR<n>ADDR contains an additional address associated with the highest priority error recorded by this record.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to 0.

Access to this field is **W1C**.

Otherwise:

Reserved, res0.

V, bit [30]

Status Register Valid.

V	Meaning
0b0	ERR<n>STATUS not valid.
0b1	ERR<n>STATUS valid. Additional syndrome has been recorded.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to 0.

Access to this field is **W1C**.

Bits [29, 25:24, 23, 19]

Reserved, RAZ.

Bits [28:27]

Reserved, res0.

MV, bit [26]
When error record <n> includes additional information for an error:

Miscellaneous Registers Valid.

MV	Meaning
0b0	ERR<n>MISC<m> not valid.
0b1	The contents of the ERR<n>MISC<m> registers contain additional information for an error recorded by this record.

Note

If the ERR<n>MISC<m> registers can contain additional information for a previously recorded error, then the contents must be self-describing to software or a user. For example, certain fields might relate only to Corrected errors, and other fields only to the most recent error that was not discarded.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to 0.

Access to this field is **W1C**.

Otherwise:

Reserved, res0.

Bits [22:20]

Reserved, res0.

Bits [18:16]

Reserved, res0.

IERR, bits [15:8]

implementation defined additional error code. Used with any primary error code ERR<n>STATUS.SERR value. Further implementation defined information can be placed in the ERR<n>MISC<m> registers.

The implemented set of valid values that this field can take is implementation defined. If any value not in this set is written to this register, then the value read back from this field is unknown.

Note

This means that one or more bits of this field might be implemented as fixed read-as-zero or read-as-one values.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - the Common Fault Injection Model Extension is not implemented by the node that owns this error record or ERR<q>PFGF.SYN == 0
  - ERR<n>STATUS.V == 0
- Otherwise, access to this field is **RW**.

Bits [7:0]

Reserved, res0.

When RAS System Architecture v2 is implemented, ERR<q>FR.ED == 0b11 and ERR<q>FR.ERT == 0b01:

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	V	ERI	RES0				FHI	RES0				CRI	RES0																		
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 [63:31]

Reserved, res0.

V, bit [30]

RAS agent error status.

V	Meaning
0b0	RAS agent error status is not asserted.
0b1	RAS agent error status is asserted.

Access to this field is **RO**.

ERI, bit [29]

RAS agent Error Recovery Interrupt.

ERI	Meaning
0b0	RAS agent error recovery interrupt is not asserted.
0b1	RAS agent error recovery interrupt is asserted.

Access to this field is **RO**.

Bits [28:25]

Reserved, res0.

FHI, bit [24]

RAS agent Fault Handling Interrupt.

FHI	Meaning
0b0	RAS agent fault handling interrupt is not asserted.
0b1	RAS agent fault handling interrupt is asserted.

Access to this field is **RO**.

Bits [23:20]

Reserved, res0.

CRI, bit [19]

RAS agent criticial error interrupt.

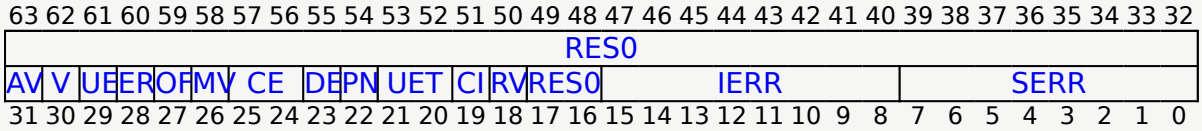
CRI	Meaning
0b0	RAS agent criticial error interrupt is not asserted.
0b1	RAS agent criticial error interrupt is asserted.

Access to this field is **RO**.

Bits [18:0]

Reserved, res0.

When RAS System Architecture v1.1 is implemented:



Bits [63:32]

Reserved, res0.

AV, bit [31]
When error record <n> includes an address associated with an error:

Address Valid.

AV	Meaning
0b0	ERR<n>ADDR not valid.
0b1	ERR<n>ADDR contains an address associated with the highest priority error recorded by this record.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to 0.

Access to this field is **W1C**.

Otherwise:

Reserved, res0.

V, bit [30]

Status Register Valid.

V	Meaning
0b0	ERR<n>STATUS not valid.
0b1	ERR<n>STATUS valid. At least one error has been recorded.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to `0`.

Access to this field is **W1C**.

UE, bit [29]

Uncorrected Error.

UE	Meaning
`0b0`	No errors have been detected, or all detected errors have been either corrected or deferred.
`0b1`	At least one detected error was not corrected and not deferred.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Otherwise, access to this field is **W1C**.

ER, bit [28]
When in-band error responses can be returned for a Deferred error:

Error Reported.

ER	Meaning
`0b0`	No in-band error response (External Abort) signaled to the Requester making the access or other transaction.

0b1	An in-band error response was signaled by the component to the Requester making the access or other transaction. This can be because any of the following are true: • The ERR<q>CTLR.UE field, or applicable one of the ERR<q>CTLR.{WUE, RUE} fields, is implemented and was 1 when an error was detected and not corrected. • The ERR<q>CTLR.{WUE, RUE, UE} fields are not implemented and the component always reports errors.

Note

An in-band error response signaled by the component might be masked and not generate any exception.

It is implementation defined whether an uncorrected error that is deferred and recorded as a Deferred error, but is not deferred to the Requester, can signal an in-band error response to the Requester, causing this field to be set to 1.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - ERR<n>STATUS.V == 0
  - ERR<n>STATUS.[DE,UE] == 0b00
- Otherwise, access to this field is **W1C**.

When in-band error responses are never be returned for a Deferred error:

Error Reported.

ER	Meaning

0b0	No in-band error response (External Abort) signaled to the Requester making the access or other transaction.
0b1	An in-band error response was signaled by the component to the Requester making the access or other transaction. This can be because any of the following are true: • The ERR<q>CTLR.UE field, or applicable one of the ERR<q>CTLR.{WUE, RUE} fields, is implemented and was 1 when an error was detected and not corrected. • The ERR<q>CTLR.{WUE, RUE, UE} fields are not implemented and the component always reports errors.

Note

An in-band error response signaled by the component might be masked and not generate any exception.

It is implementation defined whether an uncorrected error that is deferred and recorded as a Deferred error, but is not deferred to the Requester, can signal an in-band error response to the Requester, causing this field to be set to 1.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - ERR<n>STATUS.V == 0
  - ERR<n>STATUS.UE == 0
- Otherwise, access to this field is **W1C**.

Otherwise:

Reserved, res0.

OF, bit [27]

Overflow.

Indicates that multiple errors have been detected. This field is set to 1 when one of the following occurs:

- A Corrected error counter is implemented, an error is counted, and the counter overflows.
- ERR<n>STATUS.V was previously 1, a Corrected error counter is not implemented, and a Corrected error is recorded.
- ERR<n>STATUS.V was previously 1, and a type of error other than a Corrected error is recorded.

Otherwise, this field is unchanged when an error is recorded.

If a Corrected error counter is implemented, then:

- A direct write that modifies the counter overflow flag indirectly might set this field to an unknown value.
- A direct write to this field that clears this field to zero might indirectly set the counter overflow flag to an unknown value.

OF	Meaning
0b0	Since this field was last cleared to zero, no error syndrome has been discarded and, if a Corrected error counter is implemented, it has not overflowed.
0b1	Since this field was last cleared to zero, at least one error syndrome has been discarded or, if a Corrected error counter is implemented, it might have overflowed.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Otherwise, access to this field is **W1C**.

MV, bit [26]
When error record <n> includes additional information for an error:

Miscellaneous Registers Valid.

MV	Meaning
0b0	ERR<n>MISC<m> not valid.
0b1	The contents of the ERR<n>MISC<m> registers contain additional information for an error recorded by this record.

Note

If the ERR<n>MISC<m> registers can contain additional information for a previously recorded error, then the contents must be self-describing to software or a user. For example, certain fields might relate only to Corrected errors, and other fields only to the most recent error that was not discarded.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to 0.

Access to this field is **W1C**.

Otherwise:

Reserved, res0.

CE, bits [25:24]

Corrected Error.

CE	Meaning
0b00	No errors were corrected.
0b01	At least one transient error was corrected.
0b10	At least one error was corrected.
0b11	At least one persistent error was corrected.

The mechanism by which a component or node detects whether a Corrected error is transient or persistent is implementation defined. If no such mechanism is implemented, then the node sets this field to `0b10` when a corrected error is recorded.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write ones to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Otherwise, access to this field is **W1C**.

DE, bit [23]

Deferred Error.

DE	Meaning
`0b0`	No errors were deferred.
`0b1`	At least one error was not corrected and deferred.

Support for deferring errors is implementation defined.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Otherwise, access to this field is **W1C**.

PN, bit [22]

Poison.

PN	Meaning

0b0	Uncorrected error or Deferred error recorded because a corrupt value was detected, for example, by an error detection code (EDC), or Corrected error recorded.
0b1	Uncorrected error or Deferred error recorded because a poison value was detected.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - ERR<n>STATUS.V == 0
  - ERR<n>STATUS.[DE,UE] == 0b00
- Otherwise, access to this field is **W1C**.

UET, bits [21:20]

Uncorrected Error Type. Describes the state of the component after detecting or consuming an Uncorrected error.

UET	Meaning
0b00	Uncorrected error, Uncontainable error (UC).
0b01	Uncorrected error, Unrecoverable error (UEU).
0b10	Uncorrected error, Latent or Restartable error (UEO).
0b11	Uncorrected error, Signaled or Recoverable error (UER).

UER can mean either Signaled or Recoverable error, and UEO can mean either Latent or Restartable error.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write ones to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - ERR<n>STATUS.V == 0
  - ERR<n>STATUS.UE == 0
- Otherwise, access to this field is **W1C**.

CI, bit [19]

Critical Error. Indicates whether a critical error condition has been recorded.

CI	Meaning
0b0	No critical error condition.
0b1	Critical error condition.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Otherwise, access to this field is **W1C**.

RV, bit [18]
When RAS System Architecture v2 is implemented:

Reset Valid. When ERR<n>STATUS.V is 1, indicating the error record is valid, this field indicates whether the error was recorded before or after the most recent Error Recovery reset.

RV	Meaning
0b0	If the error record is valid then it was recorded after the last Error Recovery reset.
0b1	If the error record is valid then it was recorded before the last Error Recovery reset.

This field is set to 0 when an error is recorded.

The reset behavior of this field is:

- On an Error recovery reset, this field resets to 1.

Access to this field is **W1C**.

Otherwise:

Reserved, res0.

Bits [17:16]

Reserved, res0.

IERR, bits [15:8]

implementation defined error code. Used with any primary error code ERR<n>STATUS.SERR value. Further implementation defined information can be placed in the ERR<n>MISC<m> registers.

The implemented set of valid values that this field can take is implementation defined. If any value not in this set is written to this register, then the value read back from this field is unknown.

Note

This means that one or more bits of this field might be implemented as fixed read-as-zero or read-as-one values.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - the Common Fault Injection Model Extension is not implemented by the node that owns this error record
  - ERR<n>STATUS.V == 0
- Access is **UNKNOWN/WI** if all of the following are true:
  - ERR<q>PFGF.SYN == 0
  - ERR<n>STATUS.V == 0
- Otherwise, access to this field is **RW**.

SERR, bits [7:0]

Architecturally-defined primary error code. The primary error code might be used by a fault handling agent to triage an error without requiring device-specific code. For example, to count and threshold corrected errors in software, or generate a short log entry.

SERR	Meaning

`0x00`	No error.
`0x01`	implementation defined error.
`0x02`	Data value from (non-associative) internal memory. For example, ECC from on-chip SRAM or buffer.
`0x03`	implementation defined pin. For example, **nSEI** pin.
`0x04`	Assertion failure. For example, consistency failure.
`0x05`	Error detected on internal data path. For example, parity on ALU result.
`0x06`	Data value from associative memory. For example, ECC error on cache data.
`0x07`	Address/control value from associative memory. For example, ECC error on cache tag.
`0x08`	Data value from a TLB. For example, ECC error on TLB data.
`0x09`	Address/control value from a TLB. For example, ECC error on TLB tag.
`0x0A`	Data value from producer. For example, parity error on write data bus.
`0x0B`	Address/control value from producer. For example, parity error on address bus.
`0x0C`	Data value from (non-associative) external memory. For example, ECC error in SDRAM.
`0x0D`	Illegal address (software fault). For example, access to unpopulated memory.
`0x0E`	Illegal access (software fault). For example, byte write to word register.
`0x0F`	Illegal state (software fault). For example, device not ready.
`0x10`	Internal data register. For example, parity on a SIMD&FP register. For a PE, all general-purpose, stack pointer, SIMD&FP, SVE, and SME registers are data registers.

0x11	Internal control register. For example, parity on a System register. For a PE, all registers other than general-purpose, stack pointer, SIMD&FP, SVE, and SME registers are control registers.
0x12	Error response from Completer of access. For example, error response from cache write-back.
0x13	External timeout. For example, timeout on interaction with another component.
0x14	Internal timeout. For example, timeout on interface within the component.
0x15	Deferred error from Completer not supported at Requester. For example, poisoned data received from the Completer of an access by a Requester that cannot defer the error further.
0x16	Deferred error from Requester not supported at Completer. For example, poisoned data received from the Requester of an access by a Completer that cannot defer the error further.
0x17	Deferred error from Completer passed through. For example, poisoned data received from the Completer of an access and returned to the Requester.
0x18	Deferred error from Requester passed through. For example, poisoned data received from the Requester of an access and deferred to the Completer.
0x19	Error recorded by PCIe error logs. Indicates that the component has recorded an error in a PCIe error log. This might be the PCIe device status register, AER, DVSEC, or other mechanisms defined by PCIe.
0x1A	Other internal error. For example, parity error on internal state of the component that is not covered by another primary error code.

All other values are reserved.

The implemented set of valid values that this field can take is implementation defined. If any value not in this set is written to this register, then the value read back from this field is unknown.

Note

This means that one or more bits of this field might be implemented as fixed read-as-zero or read-as-one values.

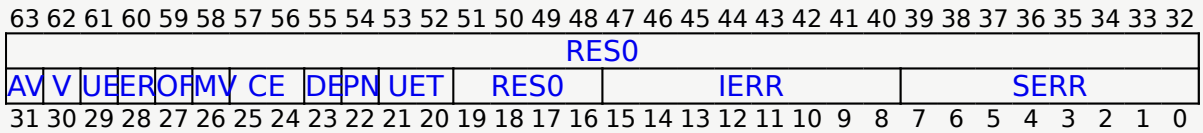
The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - the Common Fault Injection Model Extension is not implemented by the node that owns this error record
  - ERR<n>STATUS.V == 0
- Access is **UNKNOWN/WI** if all of the following are true:
  - ERR<q>PFGF.SYN == 0
  - ERR<n>STATUS.V == 0
- Otherwise, access to this field is **RW**.

Otherwise:



Bits [63:32]

Reserved, res0.

AV, bit [31]
When error record <n> includes an address associated with an error:

Address Valid.

AV	Meaning
0b0	ERR<n>ADDR not valid.
0b1	ERR<n>ADDR contains an address associated with the highest priority error recorded by this record.

The reset behavior of this field is:

- On a Cold reset:
    - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
    - Otherwise, this field resets to `0`.

Accessing this field has the following behavior:

- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.[DE,UE] == 0b00
    - ERR<n>STATUS.CE != 0b00
    - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.UE == 0
    - ERR<n>STATUS.DE != 0
    - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.UE != 0
    - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

Otherwise:

Reserved, res0.

V, bit [30]

Status Register Valid.

V	Meaning
`0b0`	ERR<n>STATUS not valid.
`0b1`	ERR<n>STATUS valid. At least one error has been recorded.

The reset behavior of this field is:

- On a Cold reset:
    - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
    - Otherwise, this field resets to `0`.

Accessing this field has the following behavior:

- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.CE != 0b00

        - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write
    - Access is **RO** if all of the following are true:
        - ERR<n>STATUS.DE != 0
        - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
    - Access is **RO** if all of the following are true:
        - ERR<n>STATUS.UE != 0
        - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
    - Otherwise, access to this field is **W1C**.

UE, bit [29]

Uncorrected Error.

UE	Meaning
0b0	No errors have been detected, or all detected errors have been either corrected or deferred.
0b1	At least one detected error was not corrected and not deferred.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.OF == 1
    - ERR<n>STATUS.OF is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

ER, bit [28]
When in-band error responses can be returned for a Deferred error:

Error Reported.

ER	Meaning
0b0	No in-band error response (External Abort) signaled to the Requester making the access or other transaction.

0b1	An in-band error response was signaled by the component to the Requester making the access or other transaction. This can be because any of the following are true: • The ERR<q>CTLR.UE field, or applicable one of the ERR<q>CTLR.{WUE, RUE} fields, is implemented and was 1 when an error was detected and not corrected. • The ERR<q>CTLR.{WUE, RUE, UE} fields are not implemented and the component always reports errors.

If this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero, when any of:

- Clearing ERR<n>STATUS.V to 0.
- Clearing both ERR<n>STATUS.{DE, UE} to 0.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - ERR<n>STATUS.V == 0
  - ERR<n>STATUS.[DE,UE] == 0b00
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE != 0
  - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE == 0
  - ERR<n>STATUS.DE != 0
  - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

When in-band error responses are never be returned for a Deferred error:

Error Reported.

ER	Meaning

0b0	No in-band error response (External Abort) signaled to the Requester making the access or other transaction.
0b1	An in-band error response was signaled by the component to the Requester making the access or other transaction. This can be because any of the following are true: • The ERR<q>CTLR.UE field, or applicable one of the ERR<q>CTLR.{WUE, RUE} fields, is implemented and was 1 when an error was detected and not corrected. • The ERR<q>CTLR.{WUE, RUE, UE} fields are not implemented and the component always reports errors.

If this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero, when any of:

- Clearing ERR<n>STATUS.V to 0.
- Clearing ERR<n>STATUS.UE to 0.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - ERR<n>STATUS.V == 0
  - ERR<n>STATUS.UE == 0
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE != 0
  - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

Otherwise:

Reserved, res0.

OF, bit [27]

Overflow.

Indicates that multiple errors have been detected. This field is set to 1 when one of the following occurs:

- An Uncorrected error is detected and ERR<n>STATUS.UE == 1.
- A Deferred error is detected, ERR<n>STATUS.UE == 0 and ERR<n>STATUS.DE == 1.
- A Corrected error is detected, no Corrected error counter is implemented, ERR<n>STATUS.UE == 0, ERR<n>STATUS.DE == 0, and ERR<n>STATUS.CE != `0b00`. ERR<n>STATUS.CE might be updated for the new Corrected error.
- A Corrected error counter is implemented, ERR<n>STATUS.UE == 0, ERR<n>STATUS.DE == 0, and the counter overflows.

It is implementation defined whether this field is set to 1 when one of the following occurs:

- A Deferred error is detected and ERR<n>STATUS.UE == 1.
- A Corrected error is detected, no Corrected error counter is implemented, and ERR<n>STATUS.{UE, DE} != {0, 0}.
- A Corrected error counter is implemented, ERR<n>STATUS.{UE, DE} != {0, 0}, and the counter overflows.

It is implementation defined whether this field is cleared to 0 when one of the following occurs:

- An Uncorrected error is detected and ERR<n>STATUS.UE == 0.
- A Deferred error is detected, ERR<n>STATUS.UE == 0, and ERR<n>STATUS.DE == 0.
- A Corrected error is detected, ERR<n>STATUS.UE == 0, ERR<n>STATUS.DE == 0, and ERR<n>STATUS.CE == `0b00`.

The implementation defined clearing of this field might also depend on the value of the other error status fields.

If a Corrected error counter is implemented, then:

- A direct write that modifies the counter overflow flag indirectly might set this field to an unknown value.
- A direct write to this field that clears this field to 0 might indirectly set the counter overflow flag to an unknown value.

OF	Meaning

`0b0`	If ERR<n>STATUS.UE == 1, then no error syndrome for an Uncorrected error has been discarded. If ERR<n>STATUS.UE == 0 and ERR<n>STATUS.DE == 1, then no error syndrome for a Deferred error has been discarded. If ERR<n>STATUS.UE == 0, ERR<n>STATUS.DE == 0, and a Corrected error counter is implemented, then the counter has not overflowed. If ERR<n>STATUS.UE == 0, ERR<n>STATUS.DE == 0, ERR<n>STATUS.CE != `0b00`, and no Corrected error counter is implemented, then no error syndrome for a Corrected error has been discarded. **Note** This field might have been set to 1 when an error syndrome was discarded and later cleared to 0 when a higher priority syndrome was recorded.
`0b1`	At least one error syndrome has been discarded or, if a Corrected error counter is implemented, it might have overflowed.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Otherwise, access to this field is **W1C**.

MV, bit [26]
When error record <n> includes additional information for an error:

Miscellaneous Registers Valid.

MV	Meaning
`0b0`	ERR<n>MISC<m> not valid.
`0b1`	The contents of the ERR<n>MISC<m> registers contain additional information for an error recorded by this record.

Note

If the ERR<n>MISC<m> registers can contain additional information for a previously recorded error, then the contents must be self-describing to software or a user. For example, certain fields might relate only to Corrected errors, and other fields only to the most recent error that was not discarded.

The reset behavior of this field is:

- On a Cold reset:
  - When RAS System Architecture v2 is implemented and ERR<q>FR.SRV == 1, this field resets to an architecturally unknown value.
  - Otherwise, this field resets to `0`.

Accessing this field has the following behavior:

- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.[DE,UE] == 0b00
  - ERR<n>STATUS.CE != 0b00
  - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE == 0
  - ERR<n>STATUS.DE != 0
  - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE != 0
  - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

Otherwise:

Reserved, res0.

CE, bits [25:24]

Corrected Error.

CE	Meaning
0b00	No errors were corrected.
0b01	At least one transient error was corrected.
0b10	At least one error was corrected.
0b11	At least one persistent error was corrected.

The mechanism by which a component or node detects whether a Corrected error is transient or persistent is implementation defined. If no such mechanism is implemented, then the node sets this field to 0b10 when a corrected error is recorded.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write ones to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.OF == 1
    - ERR<n>STATUS.OF is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

DE, bit [23]

Deferred Error.

DE	Meaning
0b0	No errors were deferred.
0b1	At least one error was not corrected and deferred.

Support for deferring errors is implementation defined.

When clearing ERR<n>STATUS.V to 0, if this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When ERR<n>STATUS.V == 0, access to this field is **UNKNOWN/WI**.
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.OF == 1
    - ERR<n>STATUS.OF is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

PN, bit [22]

Poison.

PN	Meaning
0b0	Uncorrected error or Deferred error recorded because a corrupt value was detected, for example, by an error detection code (EDC), or Corrected error recorded.
0b1	Uncorrected error or Deferred error recorded because a poison value was detected.

If this field is nonzero, then Arm recommends that software write 1 to this field to clear this field to zero, when any of:

- Clearing ERR<n>STATUS.V to 0.
- Clearing both ERR<n>STATUS.{DE, UE} to 0.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
    - ERR<n>STATUS.V == 0
    - ERR<n>STATUS.[DE,UE] == 0b00
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.[DE,UE] == 0b00
    - ERR<n>STATUS.CE != 0b00

        - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write
    - Access is **RO** if all of the following are true:
        - ERR<n>STATUS.UE == 0
        - ERR<n>STATUS.DE != 0
        - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
    - Access is **RO** if all of the following are true:
        - ERR<n>STATUS.UE != 0
        - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
    - Otherwise, access to this field is **W1C**.

UET, bits [21:20]

Uncorrected Error Type. Describes the state of the component after detecting or consuming an Uncorrected error.

UET	Meaning
`0b00`	Uncorrected error, Uncontainable error (UC).
`0b01`	Uncorrected error, Unrecoverable error (UEU).
`0b10`	Uncorrected error, Latent or Restartable error (UEO).
`0b11`	Uncorrected error, Signaled or Recoverable error (UER).

UER can mean either Signaled or Recoverable error, and UEO can mean either Latent or Restartable error.

If this field is nonzero, then Arm recommends that software write ones to this field to clear this field to zero, when any of:

- Clearing ERR<n>STATUS.V to 0.
- Clearing ERR<n>STATUS.UE to 0.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
    - ERR<n>STATUS.V == 0
    - ERR<n>STATUS.UE == 0
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.[DE,UE] == 0b00
    - ERR<n>STATUS.CE != 0b00
    - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write

- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.UE == 0
    - ERR<n>STATUS.DE != 0
    - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.UE != 0
    - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **W1C**.

Bits [19:16]

Reserved, res0.

IERR, bits [15:8]

implementation defined error code. Used with any primary error code ERR<n>STATUS.SERR value. Further implementation defined information can be placed in the ERR<n>MISC<m> registers.

The implemented set of valid values that this field can take is implementation defined. If any value not in this set is written to this register, then the value read back from this field is unknown.

Note

This means that one or more bits of this field might be implemented as fixed read-as-zero or read-as-one values.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
    - the Common Fault Injection Model Extension is not implemented by the node that owns this error record
    - ERR<n>STATUS.V == 0
- Access is **UNKNOWN/WI** if all of the following are true:
    - ERR<q>PFGF.SYN == 0
    - ERR<n>STATUS.V == 0
- Access is **RO** if all of the following are true:
    - ERR<n>STATUS.[DE,UE] == 0b00
    - ERR<n>STATUS.CE != 0b00
    - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write

- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE == 0
  - ERR<n>STATUS.DE != 0
  - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.UE != 0
  - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
- Otherwise, access to this field is **RW**.

SERR, bits [7:0]

Architecturally-defined primary error code. The primary error code might be used by a fault handling agent to triage an error without requiring device-specific code. For example, to count and threshold corrected errors in software, or generate a short log entry.

SERR	Meaning
0x00	No error.
0x01	implementation defined error.
0x02	Data value from (non-associative) internal memory. For example, ECC from on-chip SRAM or buffer.
0x03	implementation defined pin. For example, **nSEI** pin.
0x04	Assertion failure. For example, consistency failure.
0x05	Error detected on internal data path. For example, parity on ALU result.
0x06	Data value from associative memory. For example, ECC error on cache data.
0x07	Address/control value from associative memory. For example, ECC error on cache tag.
0x08	Data value from a TLB. For example, ECC error on TLB data.
0x09	Address/control value from a TLB. For example, ECC error on TLB tag.
0x0A	Data value from producer. For example, parity error on write data bus.
0x0B	Address/control value from producer. For example, parity error on address bus.

`0x0C`	Data value from (non-associative) external memory. For example, ECC error in SDRAM.
`0x0D`	Illegal address (software fault). For example, access to unpopulated memory.
`0x0E`	Illegal access (software fault). For example, byte write to word register.
`0x0F`	Illegal state (software fault). For example, device not ready.
`0x10`	Internal data register. For example, parity on a SIMD&FP register. For a PE, all general-purpose, stack pointer, SIMD&FP, SVE, and SME registers are data registers.
`0x11`	Internal control register. For example, parity on a System register. For a PE, all registers other than general-purpose, stack pointer, SIMD&FP, SVE, and SME registers are control registers.
`0x12`	Error response from Completer of access. For example, error response from cache write-back.
`0x13`	External timeout. For example, timeout on interaction with another component.
`0x14`	Internal timeout. For example, timeout on interface within the component.
`0x15`	Deferred error from Completer not supported at Requester. For example, poisoned data received from the Completer of an access by a Requester that cannot defer the error further.
`0x16`	Deferred error from Requester not supported at Completer. For example, poisoned data received from the Requester of an access by a Completer that cannot defer the error further.

0x17	Deferred error from Completer passed through. For example, poisoned data received from the Completer of an access and returned to the Requester.
0x18	Deferred error from Requester passed through. For example, poisoned data received from the Requester of an access and deferred to the Completer.
0x19	Error recorded by PCIe error logs. Indicates that the component has recorded an error in a PCIe error log. This might be the PCIe device status register, AER, DVSEC, or other mechanisms defined by PCIe.
0x1A	Other internal error. For example, parity error on internal state of the component that is not covered by another primary error code.

All other values are reserved.

The implemented set of valid values that this field can take is implementation defined. If any value not in this set is written to this register, then the value read back from this field is unknown.

Note

This means that one or more bits of this field might be implemented as fixed read-as-zero or read-as-one values.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - the Common Fault Injection Model Extension is not implemented by the node that owns this error record
  - ERR<n>STATUS.V == 0
- Access is **UNKNOWN/WI** if all of the following are true:
  - ERR<q>PFGF.SYN == 0
  - ERR<n>STATUS.V == 0
- Access is **RO** if all of the following are true:
  - ERR<n>STATUS.DE == 0

        - ERR<n>STATUS.UE == 0
        - ERR<n>STATUS.CE != 0b00
        - ERR<n>STATUS.CE is not being cleared to 0b00 in the same write
    - Access is **RO** if all of the following are true:
        - ERR<n>STATUS.UE == 0
        - ERR<n>STATUS.DE != 0
        - ERR<n>STATUS.DE is not being cleared to 0b0 in the same write
    - Access is **RO** if all of the following are true:
        - ERR<n>STATUS.UE != 0
        - ERR<n>STATUS.UE is not being cleared to 0b0 in the same write
    - Otherwise, access to this field is **RW**.

Accessing ERR<n>STATUS

ERR<n>STATUS.{AV, V, UE, ER, OF, MV, CE, DE, PN, UET, CI} are write-one-to-clear (W1C) fields, meaning writes of zero are ignored, and a write of one or all-ones to the field clears the field to zero. ERR<n>STATUS.{IERR, SERR} are read/write (RW) fields, although the set of implemented valid values is implementation defined. See also ERR<n>PFGF.SYN.

After reading ERR<n>STATUS, software must clear the valid fields in the register to allow new errors to be recorded. However, between reading the register and clearing the valid fields, a new error might have overwritten the register. To prevent this error being lost by software, the register prevents updates to fields that might have been updated by a new error.

When RAS System Architecture v1.0 is implemented:

- Writes to ERR<n>STATUS.{UE, DE, CE} are ignored if ERR<n>STATUS.OF is 1 and is not being cleared to 0.
- Writes to ERR<n>STATUS.V are ignored if any of ERR<n>STATUS.{UE, DE, CE} are nonzero and are not being cleared to zero.
- Writes to ERR<n>STATUS.{AV, MV} and the ERR<n>STATUS.{ER, PN, UET, IERR, SERR} syndrome fields are ignored if the highest priority nonzero error status field is not being cleared to zero. The error status fields in priority order from highest to lowest, are ERR<n>STATUS.UE, ERR<n>STATUS.DE, and ERR<n>STATUS.CE.

When RAS System Architecture v1.1 is implemented, a write to the register is ignored if all of:

- Any of ERR<n>STATUS.{V, UE, OF, CE, DE} are nonzero before the write.
- The write does not clear the nonzero ERR<n>STATUS.{V, UE, OF, CE, DE} fields to zero by writing ones to the applicable field or fields.

Some of the fields in ERR<n>STATUS are also defined as unknown where certain combinations of ERR<n>STATUS.{V, DE, UE} are zero. The rules for writes to ERR<n>STATUS allow a node to implement such a field as a fixed read-only value.

For example, when RAS System Architecture v1.1 is implemented, a write to ERR<n>STATUS when ERR<n>STATUS.V is 1 results in either ERR<n>STATUS.V field being cleared to zero, or ERR<n>STATUS.V not changing. Since all fields in ERR<n>STATUS, other than ERR<n>STATUS.{AV, V, MV}, usually read as unknown values when ERR<n>STATUS.V is zero, this means those fields can be implemented as read-only if applicable.

To ensure correct and portable operation, when software is clearing the valid fields in the register to allow new errors to be recorded, Arm recommends that software performs the following sequence of operations in order:

- Read ERR<n>STATUS and determine which fields need to be cleared to zero.
- In a single write to ERR<n>STATUS:
    - Write ones to all the W1C fields that are nonzero in the read value.
    - Write zero to all the W1C fields that are zero in the read value.
    - Write zero to all the RW fields.
- Read back ERR<n>STATUS after the write to confirm no new fault has been recorded.

Otherwise, these fields might not have the correct value when a new fault is recorded.

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

Component	Offset	Instance
RAS	`0x010` + (64 * n)	ERR<n>STATUS

This interface is accessible as follows:

- When ERR<n>STATUS.V != 0, ERR<n>STATUS.V is not being cleared to 0b0 in the same write and RAS System Architecture v1.1 is implemented, accesses to this register are **RO**.
- When ERR<n>STATUS.UE != 0, ERR<n>STATUS.UE is not being cleared to 0b0 in the same write and RAS System Architecture v1.1 is implemented, accesses to this register are **RO**.
- When ERR<n>STATUS.OF != 0, ERR<n>STATUS.OF is not being cleared to 0b0 in the same write and RAS System Architecture v1.1 is implemented, accesses to this register are **RO**.

- When ERR<n>STATUS.CE != 0b00, ERR<n>STATUS.CE is not being cleared to 0b00 in the same write and RAS System Architecture v1.1 is implemented, accesses to this register are **RO**.
- When ERR<n>STATUS.DE != 0, ERR<n>STATUS.DE is not being cleared to 0b0 in the same write and RAS System Architecture v1.1 is implemented, accesses to this register are **RO**.
- Otherwise, accesses to this register are **RW**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94