EDPRSR, External Debug Processor Status Register

The EDPRSR characteristics are:

Purpose

Holds information about the reset and powerdown state of the PE.

Configuration

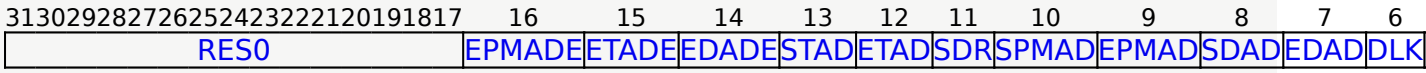
When FEAT_DoPD is implemented, EDPRSR is in the Core power domain. Otherwise, EDPRSR contains fields that are in the Core power domain and fields that are in the Debug power domain.

If FEAT_DoPD is implemented then all fields in this register are in the Core power domain.

Attributes

EDPRSR is a 32-bit register.

Field descriptions



Bits [31:17]

Reserved, res0.

EPMADE, bit [16]
When FEAT_RME is implemented and FEAT_PMUv3_EXT is implemented:

External Performance Monitors Access Disable Extended Status. Together with EDPRSR.EPMAD, reports whether access to Performance Monitor registers by an external debugger is permitted.

For a description of the values derived by evaluating EPMAD and EPMADE together, see EDPRSR.EPMAD.

Otherwise:

Reserved, res0.

ETADE, bit [15]
When FEAT_RME is implemented, FEAT_TRC_EXT is implemented and FEAT_TRBE is implemented:

External Trace Access Disable Extended Status. Together with EDPRSR.ETAD, reports whether access to trace unit registers by an external debugger is permitted.

For a description of the values derived by evaluating ETAD and ETADE together, see EDPRSR.ETAD.

Otherwise:

Reserved, res0.

EDADE, bit [14]
When FEAT_RME is implemented:

External Debug Access Disable Extended Status. Together with EDPRSR.EDAD, reports whether access to breakpoint registers, watchpoint registers, and OSLAR_EL1 by an external debugger is permitted.

For a description of the values derived by evaluating EDAD and EDADE together, see EDPRSR.EDAD.

Otherwise:

Reserved, res0.

STAD, bit [13]
When FEAT_TRC_EXT is implemented and FEAT_TRBE is implemented:

Sticky ETAD error. Set to 1 when a Non-secure external debug interface access to an external trace register returns an error because `AllowExternalTraceAccess`() == FALSE for the access.

STAD	Meaning
`0b0`	Since EDPRSR was last read, no external accesses to the trace unit registers have failed because `AllowExternalTraceAccess`() was FALSE for the access.

0b1	Since EDPRSR was last read, at least one external access to the trace unit registers has failed because AllowExternalTraceAccess() was FALSE for the access.

If IsCorePowered() == TRUE, the Core power domain is powered up, then, following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or DoubleLockStatus() == FALSE then this bit clears to 0.
- If FEAT_DoubleLock is implemented and DoubleLockStatus() == TRUE then it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This bit is in the Core power domain.

Note

If FEAT_DoPD is implemented, FEAT_DoubleLock is not implemented.

The reset behavior of this field is:

- On a Cold reset, this field resets to 0.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - DoubleLockStatus()
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RC/WI**.

Otherwise:

Reserved, res0.

ETAD, bit [12]
When FEAT_RME is implemented, FEAT_TRC_EXT is implemented and FEAT_TRBE is implemented:

External Trace Access Disable Status. Together with EDPRSR.ETADE, reports whether access to trace unit registers by an external debugger is permitted.

ETADE	ETAD	Meaning
0b0	0b0	Access to trace unit registers by an external debugger is permitted.
0b0	0b1	Root and Secure access to trace unit registers by an external debugger is permitted. Realm and Non-secure access to trace unit registers by an external debugger is not permitted.
0b1	0b0	Root and Realm access to trace unit registers by an external debugger is permitted. Secure and Non-secure access to trace unit registers by an external debugger is not permitted.
0b1	0b1	Root access to trace unit registers by an external debugger is permitted. Secure, Non-secure, and Realm access to trace unit registers by an external debugger is not permitted.

When FEAT_TRC_EXT is implemented and FEAT_TRBE is implemented:

External Trace Access Disable status.

ETAD	Meaning
0b0	External Non-secure trace unit accesses enabled. `AllowExternalTraceAccess`() == TRUE for a Non-secure access.
0b1	External Non-secure trace unit accesses disabled. `AllowExternalTraceAccess`() == FALSE for a Non-secure access.

This bit is in the Core power domain.

Note

If FEAT_DoPD is implemented, FEAT_DoubleLock is not implemented.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - DoubleLockStatus()
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

Otherwise:

Reserved, res0.

SDR, bit [11]

Sticky Debug Restart. Set to 1 when the PE exits Debug state.

Permitted values are:

SDR	Meaning
`0b0`	The PE has not restarted since EDPRSR was last read.
`0b1`	The PE has restarted since EDPRSR was last read.

Note

If a reset occurs when the PE is in Debug state, the PE exits Debug state. SDR is unknown on Warm reset, meaning a debugger must also use the SR bit to determine whether the PE has left Debug state.

If the Core power domain is powered up, then following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or `DoubleLockStatus()` == FALSE this bit clears to 0.
- If FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Warm 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:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- When SoftwareLockStatus(), access to this field is **RO**.
- Otherwise, access to this field is **RC/WI**.

SPMAD, bit [10]
When FEAT_Debugv8p4 is implemented and FEAT_PMUv3_EXT is implemented:

Sticky EPMAD error. Set to 1 if an external debug interface access to a Performance Monitors register returns an error because `AllowExternalPMUAccess()` == FALSE.

Permitted values are:

SPMAD	Meaning
`0b0`	No Non-secure external debug interface accesses to the external Performance Monitors registers have failed because `AllowExternalPMUAccess()` == FALSE for the access since EDPRSR was last read.
`0b1`	At least one Non-secure external debug interface access to the external Performance Monitors register has failed and returned an error because `AllowExternalPMUAccess()` == FALSE for the access since EDPRSR was last read.

If the Core power domain is powered up, then following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or `DoubleLockStatus()` == FALSE, this bit clears to 0.
- If FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Cold reset, this field resets to `0`.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- When SoftwareLockStatus(), access to this field is **RO**.
- Otherwise, access to this field is **RC/WI**.

When FEAT_PMUv3_EXT is implemented:

Sticky EPMAD error.

SPMAD	Meaning
`0b0`	No external debug interface accesses to the Performance Monitors registers have failed because `AllowExternalPMUAccess()` == FALSE since EDPRSR was last read.
`0b1`	At least one external debug interface access to the Performance Monitors registers has failed and returned an error because `AllowExternalPMUAccess()` == FALSE since EDPRSR was last read.

If the Core power domain is powered up, then, following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or `DoubleLockStatus()` == FALSE, this bit clears to 0.
- If FEAT_DoubleLock is implemented, and `DoubleLockStatus()` == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Cold reset, this field resets to `0`.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - OSLockStatus()

        - DoubleLockStatus()
        - EDPRSR.R == 1
    - When SoftwareLockStatus(), access to this field is **RO**.
    - Otherwise, access to this field is **RC/WI**.

Otherwise:

Reserved, res0.

EPMAD, bit [9]
When FEAT_RME is implemented and FEAT_PMUv3_EXT is implemented:

External Performance Monitors Access Disable Status. Together with EDPRSR.EPMADE, reports whether access to Performance Monitor registers by an external debugger is permitted.

EPMADE	EPMAD	Meaning
0b0	0b0	Access to Performance Monitor registers by an external debugger is permitted.
0b0	0b1	Root and Secure access to Performance Monitor registers by an external debugger is permitted. Realm and Non-secure access to Performance Monitor registers by an external debugger is not permitted.

EPMADE	EPMAD	Meaning
0b1	0b0	Root and Realm access to Performance Monitor registers by an external debugger is permitted. Secure and Non-secure access to Performance Monitor registers by an external debugger is not permitted.
0b1	0b1	Root access to Performance Monitor registers by an external debugger is permitted. Secure, Non-secure, and Realm access to Performance Monitor registers by an external debugger is not permitted.

When FEAT_Debugv8p4 is implemented and FEAT_PMUv3_EXT is implemented:

External Performance Monitors Non-secure Access Disable status.

EPMAD	Meaning
0b0	External Non-secure Performance Monitors access enabled. `AllowExternalPMUAccess()` == TRUE for a Non-secure access.
0b1	External Non-secure Performance Monitors access disabled. `AllowExternalPMUAccess()` == FALSE for a Non-secure access.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

When FEAT_PMUv3_EXT is implemented and FEAT_Debugv8p4 is not implemented:

External Performance Monitors access disable status.

EPMAD	Meaning
0b0	External Performance Monitors access enabled. `AllowExternalPMUAccess()` == TRUE.
0b1	External Performance Monitors access disabled. `AllowExternalPMUAccess()` == FALSE.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - OSLockStatus()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

Otherwise:

Reserved, res0.

SDAD, bit [8]
When FEAT_Debugv8p4 is implemented:

Sticky EDAD error. Set to 1 if an external debug interface access to a debug register returns an error because `AllowExternalDebugAccess()` == FALSE.

SDAD	Meaning

0b0	No Non-secure external debug interface accesses to the debug registers have failed because AllowExternalDebugAccess() == FALSE for the access since EDPRSR was last read.
0b1	At least one Non-secure external debug interface access to the debug registers has failed and returned an error because AllowExternalDebugAccess() == FALSE for the access since EDPRSR was last read.

If the Core power domain is powered up, then, following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or DoubleLockStatus() == FALSE this bit clears to 0.
- If FEAT_DoubleLock is implemented and DoubleLockStatus() == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Cold reset, this field resets to 0.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

Otherwise:

Sticky EDAD error. Set to 1 if an external debug interface access to a debug register returns an error because AllowExternalDebugAccess() == FALSE.

SDAD	Meaning
0b0	No external debug interface accesses to the debug registers have failed because AllowExternalDebugAccess() == FALSE since EDPRSR was last read.

0b1	At least one external debug interface access to the debug registers has failed and returned an error because AllowExternalDebugAccess() == FALSE since EDPRSR was last read.

If the Core power domain is powered up, then, following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or DoubleLockStatus() == FALSE this bit clears to 0.
- If FEAT_DoubleLock is implemented and DoubleLockStatus() == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This bit is unknown on reads if OSLockStatus() == TRUE and external debug writes to OSLAR_EL1 do not return an error when AllowExternalDebugAccess() == FALSE.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Cold reset, this field resets to 0.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

EDAD, bit [7]
When FEAT_RME is implemented:

External Debug Access Disable Status. Together with EDPRSR.EDADE, reports whether access to breakpoint registers, watchpoint registers, and OSLAR_EL1 by an external debugger is permitted.

EDADE	EDAD	Meaning
0b0	0b0	Access to Debug registers by an external debugger is permitted.

EDADE	EDAD	Meaning
0b0	0b1	Root and Secure access to Debug registers by an external debugger is permitted. Realm and Non-secure access to Debug registers by an external debugger is not permitted.
0b1	0b0	Root and Realm access to Debug registers by an external debugger is permitted. Secure and Non-secure access to Debug registers by an external debugger is not permitted.
0b1	0b1	Root access to Debug registers by an external debugger is permitted. Secure, Non-secure, and Realm access to Debug registers by an external debugger is not permitted.

When FEAT_Debugv8p4 is implemented:

External Debug Access Disable status.

EDAD	Meaning
0b0	External Non-secure access to breakpoint registers, watchpoint registers, and OSLAR_EL1 enabled. `AllowExternalDebugAccess()` == TRUE for a Non-secure access.
0b1	External Non-secure access to breakpoint registers, watchpoint registers, and OSLAR_EL1 disabled. `AllowExternalDebugAccess()` == FALSE for a Non-secure access.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

When FEAT_Debugv8p2 is implemented:

External Debug Access Disable status.

EDAD	Meaning
0b0	External access to breakpoint registers, watchpoint registers, and OSLAR_EL1 enabled. `AllowExternalDebugAccess()` == TRUE.
0b1	External access to breakpoint registers, watchpoint registers, and OSLAR_EL1 disabled. `AllowExternalDebugAccess()` == FALSE.

This bit is not valid and reads unknown if `OSLockStatus()` == TRUE and external debug writes to OSLAR_EL1 do not return an error when `AllowExternalDebugAccess()` == FALSE.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

Otherwise:

External Debug Access Disable status.

EDAD	Meaning
0b0	External access to breakpoint registers, watchpoint registers, and OSLAR_EL1 enabled. `AllowExternalDebugAccess()` == TRUE.

0b1	External access to breakpoint registers, watchpoint registers disabled. It is implementation defined whether accesses to OSLAR_EL1 are enabled or disabled. `AllowExternalDebugAccess()` == FALSE.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

DLK, bit [6]
When FEAT_Debugv8p4 is implemented:

This field is res0.

When FEAT_Debugv8p2 is implemented and FEAT_DoubleLock is implemented:

Double Lock.

From Armv8.2, this field is deprecated.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **RAZ/WI** if all of the following are true:
  - IsCorePowered()
  - !DoubleLockStatus()
- Otherwise, access to this field is **UNKNOWN/WI**.

When FEAT_DoubleLock is implemented:

Double Lock.

This field returns the result of the pseudocode function `DoubleLockStatus()`.

If the Core power domain is powered up and `DoubleLockStatus()` == TRUE, it is implementation defined whether:

- EDPRSR.PU reads as 1, EDPRSR.DLK reads as 1, and EDPRSR.SPD is unknown.

- EDPRSR.PU reads as 0, EDPRSR.DLK is unknown, and EDPRSR.SPD reads as 0.

This field is in the Core power domain.

DLK	Meaning
0b0	`DoubleLockStatus()` returns FALSE.
0b1	`DoubleLockStatus()` returns TRUE and the Core power domain is powered up.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - FEAT_DoPD is not implemented
  - !IsCorePowered()
- Otherwise, access to this field is **RO**.

Otherwise:

Reserved, res0.

OSLK, bit [5]

OS Lock status bit.

A read of this bit returns the value of OSLSR_EL1.OSLK.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - FEAT_DoPD is not implemented
  - !IsCorePowered()
  - DoubleLockStatus()
  - EDPRSR.R == 1
- Otherwise, access to this field is **RO**.

HALTED, bit [4]

Halted status bit.

HALTED	Meaning
0b0	PE is in Non-debug state.
0b1	PE is in Debug state.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if all of the following are true:
  - FEAT_DoPD is not implemented
  - !IsCorePowered()
- Otherwise, access to this field is **RO**.

SR, bit [3]

Sticky core Reset status bit.

Permitted values are:

SR	Meaning
`0b0`	The non-debug logic of the PE is not in reset state and has not been reset since the last time EDPRSR was read.
`0b1`	The non-debug logic of the PE is in reset state or has been reset since the last time EDPRSR was read.

If EDPRSR.PU reads as 1 and EDPRSR.R reads as 0, which means that the Core power domain is in a powerup state and that the non-debug logic of the PE is not in reset state, then following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or `DoubleLockStatus()` == FALSE this bit clears to 0.
- If FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Warm reset, this field resets to `1`.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
- When SoftwareLockStatus(), access to this field is **RO**.
- Otherwise, access to this field is **RC/WI**.

R, bit [2]

PE Reset status bit.

Permitted values are:

R	Meaning
0b0	The non-debug logic of the PE is not in reset state.
0b1	The non-debug logic of the PE is in reset state.

If FEAT_DoubleLock is implemented, the PE is in reset state, and the PE entered reset state with the OS Double Lock locked this bit has a constrained unpredictable value. For more information, see 'EDPRSR.{DLK, R} and reset state'.

This field is in the Core power domain.

Accessing this field has the following behavior:

- Access is **UNKNOWN/WI** if any of the following are true:
  - FEAT_DoPD is not implemented and !IsCorePowered()
  - DoubleLockStatus()
- Otherwise, access to this field is **RO**.

SPD, bit [1]

Sticky core Powerdown status bit.

If FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, then:

- If FEAT_Debugv8p2 is implemented, this bit reads as 0.
- If FEAT_Debugv8p2 is not implemented, this bit might read as 0 or 1.

For more information, see 'EDPRSR.{DLK, SPD, PU} and the Core power domain'.

SPD	Meaning
0b0	If EDPRSR.PU is 0, it is not known whether the state of the debug registers in the Core power domain is lost. If EDPRSR.PU is 1, the state of the debug registers in the Core power domain has not been lost.
0b1	The state of the debug registers in the Core power domain has been lost.

If the Core power domain is powered up, then, following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or `DoubleLockStatus()` == FALSE this bit clears to 0.

- If FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, it is constrained unpredictable whether this bit clears to 0 or is unchanged.

When FEAT_DoPD is not implemented and the Core power domain is in either retention or powerdown state, the value of EDPRSR.SPD is implementation defined. For more information, see 'EDPRSR.SPD when the Core domain is in either retention or powerdown state'.

EDPRSR.{DLK, SPD, PU} describe whether registers in the Core power domain can be accessed, and whether their state has been lost since the last time the register was read. For more information, see 'EDPRSR.{DLK, SPD, PU} and the Core power domain'.

This field is in the Core power domain.

The reset behavior of this field is:

- On a Cold reset, this field resets to `1`.

Accessing this field has the following behavior:

- Access is **RAZ/WI** if all of the following are true:
  - FEAT_DoPD is not implemented
  - !IsCorePowered()
- Access is **UNKNOWN/WI** if all of the following are true:
  - IsCorePowered()
  - DoubleLockStatus()
- Otherwise, access to this field is **RO**.

PU, bit [0]
When FEAT_DoPD is implemented:

Core powerup status bit.

Access to this field is **RAO/WI**.

When FEAT_Debugv8p2 is implemented:

Core Powerup status bit. Indicates whether the debug registers in the Core power domain can be accessed.

PU	Meaning
`0b0`	Either the Core power domain is in a low-power or powerdown state, or FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, meaning the debug registers in the Core power domain cannot be accessed.

0b1	The Core power domain is in a powerup state, and either FEAT_DoubleLock is not implemented or DoubleLockStatus() == FALSE, meaning the debug registers in the Core power domain can be accessed.

If FEAT_DoubleLock is implemented, the PE is in reset state, and the PE entered reset state with the OS Double Lock locked this bit has a constrained unpredictable value. For more information, see 'EDPRSR.{DLK, R} and reset state'

EDPRSR.{DLK, SPD, PU} describe whether registers in the Core power domain can be accessed, and whether their state has been lost since the last time the register was read. For more information, see 'EDPRSR.{DLK, SPD, PU} and the Core power domain'

Access to this field is **RO**.

Otherwise:

Core Powerup status bit. Indicates whether the debug registers in the Core power domain can be accessed.

When the Core power domain is powered-up and DoubleLockStatus() == TRUE, then the value of EDPRSR.PU is implementation defined. See the description of the DLK bit for more information.

Otherwise, permitted values are:

PU	Meaning
0b0	Core power domain is in a low-power or powerdown state where the debug registers in the Core power domain cannot be accessed.
0b1	Core power domain is in a powerup state where the debug registers in the Core power domain can be accessed.

If FEAT_DoubleLock is implemented, the Core power domain is powered up, and DoubleLockStatus() == TRUE, it is implementation defined whether this bit reads as 0 or 1.

If FEAT_DoubleLock is implemented, the PE is in reset state, and the PE entered reset state with the OS Double Lock locked this bit has a constrained unpredictable value. For more information see 'EDPRSR.{DLK, R} and reset state'

EDPRSR.{DLK, SPD, PU} describe whether registers in the Core power domain can be accessed, and whether their state has been lost since the last time the register was read. For more information, see 'EDPRSR.{DLK, SPD, PU} and the Core power domain'.

Access to this field is **RO**.

Accessing EDPRSR

On permitted accesses to the register, other access controls affect the behavior of some fields. See the field descriptions for more information.

If the Core power domain is powered up (EDPRSR.PU == 1), then following a read of EDPRSR:

- If FEAT_DoubleLock is not implemented or `DoubleLockStatus()` == FALSE, then:
  - EDPRSR.{SDR, SPMAD, SDAD, SPD} are cleared to 0.
  - EDPRSR.SR is cleared to 0 if the non-debug logic of the PE is not in reset state (EDPRSR.R == 0).
- If FEAT_DoubleLock is implemented and `DoubleLockStatus()` == TRUE, it is constrained unpredictable whether or not this clearing occurs.

If FEAT_DoPD is not implemented and the Core power domain is powered down (EDPRSR.PU == 0), then:

- EDPRSR.{SDR, SPMAD, SDAD, SR} are all unknown, and are either reset or restored on being powered up.
- EDPRSR.SPD is not cleared following a read of EDPRSR. See the SPD bit description for more information.

The clearing of bits is an indirect write to EDPRSR.

EDPRSR can be accessed through the external debug interface:

Component	Offset	Instance
Debug	`0x314`	EDPRSR

This interface is accessible as follows:

- When FEAT_DoPD is not implemented or IsCorePowered(), accesses to this register are **RO**.
- Otherwise, accesses to this register generate an error response.