

## PFAR\_EL2, Physical Fault Address Register (EL2)

The PFAR\_EL2 characteristics are:

### Purpose

Records the faulting physical address for a synchronous External Abort, or SError exception taken to EL2.

### Configuration

This register is present only when FEAT\_PFAR is implemented. Otherwise, direct accesses to PFAR\_EL2 are undefined.

### Attributes

PFAR\_EL2 is a 64-bit register.

### Field descriptions

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32		
NS	NSE	RES0						PA[55:52]				PA[51:48]				PA																	
PA																																	
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		

#### NS, bit [63]

When FEAT\_RME is implemented:

Together with PFAR\_EL2.NSE, reports the physical address space of the access that triggered the exception.

NSE	NS	Meaning
0b0	0b0	When Secure state is implemented, Secure. Otherwise reserved.
0b0	0b1	Non-secure.
0b1	0b0	Reserved.
0b1	0b1	Realm.

The reset behavior of this field is:

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

**When EL3 is implemented:**

Non-secure. Reports the physical address space of the access that triggered the exception.

NS	Meaning
0b0	Secure physical address space.
0b1	Non-secure physical address space.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**NSE, bit [62]****When FEAT\_RME is implemented:**

Together with PFAR\_EL2.NS, reports the physical address space of the access that triggered the exception.

For a description of the values derived by evaluating NS and NSE together, see MFAR\_EL3.NS.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**Bits [61:56]**

Reserved, res0.

**PA[55:52], bits [55:52]****When FEAT\_D128 is implemented:**

When FEAT\_D128 is implemented, extension to PFAR\_EL2.PA[47:0].

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**PA[51:48], bits [51:48]**

**When FEAT\_LPA is implemented:**

When FEAT\_LPA is implemented, extension to PFAR\_EL2.PA[47:0].

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**PA, bits [47:0]**

Physical Address. Bits [47:0] of the aborting physical address.

For implementations with fewer than 48 physical address bits, the corresponding upper bits in this field are res0.

The recorded address can be any address within the same naturally-aligned fault granule as the faulting physical address, where the size of the fault granule is implementation defined and no larger than the larger than:

- The size of the range of values permitted to be recorded in [FAR\\_EL2](#).

The reset behavior of this field is:

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

## **Accessing PFAR\_EL2**

When FEAT\_VHE is implemented, and [HCR\\_EL2](#).E2H is 1, without explicit synchronization, accesses from EL2 using the register name PFAR\_EL2 or PFAR\_EL1 are not guaranteed to be ordered with respect to accesses using the other register name.

PFAR\_EL2 is not valid and reads unknown if [ESR\\_EL2](#).PFV is recorded as 0.

Accesses to this register use the following encodings in the System register encoding space:

## MRS <Xt>, PFAR\_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b0110	0b0000	0b101

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    X[t, 64] = PFAR_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = PFAR_EL2;
```

## MSR PFAR\_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b0110	0b0000	0b101

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    PFAR_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    PFAR_EL2 = X[t, 64];
```

---

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.