

ELR_EL2, Exception Link Register (EL2)

The ELR_EL2 characteristics are:

Purpose

When taking an exception to EL2, holds the address to return to.

Configuration

AArch64 System register ELR_EL2 bits [31:0] are architecturally mapped to AArch32 System register [ELR_hyp\[31:0\]](#).

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

ELR_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
Return address																															
Return address																															
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 [63:0]

Return address.

An exception return from EL2 using AArch64 makes ELR_EL2 become unknown.

When EL2 is in AArch32 Execution state and an exception is taken from EL0, EL1, or EL2 to EL3 and AArch64 execution, the upper 32-bits of ELR_EL2 are either set to 0 or hold the same value that they did before AArch32 execution. Which option is adopted is determined by an implementation, and might vary dynamically within an implementation. Correspondingly software must regard the value as being an unknown choice between the two values.

The reset behavior of this field is:

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

Accessing ELR_EL2

When [HCR_EL2](#).E2H is 1, without explicit synchronization, access from EL2 using the mnemonic ELR_EL2 or ELR_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

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

MRS <Xt>, ELR_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b0100	0b0000	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        X[t, 64] = ELR_EL1;
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    X[t, 64] = ELR_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ELR_EL2;
```

MSR ELR_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b0100	0b0000	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if IsFeatureImplemented(FEAT_GCS) &&
        GetCurrentEXLOCKEN() && !Halted() && PSTATE.EXLOCK
        == '1' && EL2Enabled() && HCR_EL2.NV == '1' then
        EXLOCKException();
    elsif EL2Enabled() && HCR_EL2.<NV2,NV> == '11'
    then
        ELR_EL1 = X[t, 64];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
```

```

        UNDEFINED;
    elsif PSTATE.EL == EL2 then
        if IsFeatureImplemented(FEAT_GCS) &&
            GetCurrentEXLOCKEN() && !Halted() && PSTATE.EXLOCK
            == '1' then
            EXLOCKException();
        else
            ELR_EL2 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        ELR_EL2 = X[t, 64];

```

When FEAT_VHE is implemented

MRS <Xt>, ELR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0100	0b0000	0b001

```

    if PSTATE.EL == EL0 then
        UNDEFINED;
    elsif PSTATE.EL == EL1 then
        if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '011'
        then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
            X[t, 64] = NVMem[0x230];
        else
            X[t, 64] = ELR_EL1;
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' then
            X[t, 64] = ELR_EL2;
        else
            X[t, 64] = ELR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = ELR_EL1;

```

When FEAT_VHE is implemented

MSR ELR_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b0100	0b0000	0b001

```

    if PSTATE.EL == EL0 then
        UNDEFINED;
    elsif PSTATE.EL == EL1 then

```

```

        if IsFeatureImplemented(FEAT_GCS) &&
        GetCurrentEXLOCKEN() && !Halted() && PSTATE.EXLOCK
        == '1' && (HCR_EL2.NV == '0' || (EL2Enabled() &&
        HCR_EL2.<NV1,NV> == '01')) then
            EXLOCKException();
        elseif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '011' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
            NVMem[0x230] = X[t, 64];
        else
            ELR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if IsFeatureImplemented(FEAT_GCS) &&
        GetCurrentEXLOCKEN() && !Halted() && PSTATE.EXLOCK
        == '1' && HCR_EL2.E2H == '1' then
            EXLOCKException();
        elseif HCR_EL2.E2H == '1' then
            ELR_EL2 = X[t, 64];
        else
            ELR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL3 then
        ELR_EL1 = X[t, 64];

```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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