

## APDBKeyLo\_EL1, Pointer Authentication Key B for Data (bits[63:0])

The APDBKeyLo\_EL1 characteristics are:

### Purpose

Holds bits[63:0] of key B used for authentication of data pointer values.

### Note

The term APDBKey\_EL1 is used to describe the concatenation of [APDBKeyHi\\_EL1](#): [APDBKeyLo\\_EL1](#).

### Configuration

This register is present only when FEAT\_PAAuth is implemented. Otherwise, direct accesses to APDBKeyLo\_EL1 are undefined.

### Attributes

APDBKeyLo\_EL1 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
64 bit value, bits[63:0] of the 128 bit pointer authentication key value																															
64 bit value, bits[63:0] of the 128 bit pointer authentication key value																															
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]

64 bit value, bits[63:0] of the 128 bit pointer authentication key value.

The reset behavior of this field is:

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

### Accessing APDBKeyLo\_EL1

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

## MRS <Xt>, APDBKeyLo\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0010	0b010

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.APK == '0' then
        UNDEFINED;
    elsif EL2Enabled() && HCR_EL2.APK == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGTR_EL2.APDBKey == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.APK == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = APDBKeyLo_EL1;
    elsif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
        && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.APK == '0' then
            UNDEFINED;
        elsif HaveEL(EL3) && SCR_EL3.APK == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = APDBKeyLo_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = APDBKeyLo_EL1;
```

## MSR APDBKeyLo\_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0010	0b010

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

```

        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
        && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.APK == '0' then
            UNDEFINED;
        elsif EL2Enabled() && HCR_EL2.APK == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGWTR_EL2.APDBKey == '1'
then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif HaveEL(EL3) && SCR_EL3.APK == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                APDBKeyLo_EL1 = X[t, 64];
        elsif PSTATE.EL == EL2 then
            if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.APK == '0' then
                UNDEFINED;
            elsif HaveEL(EL3) && SCR_EL3.APK == '0' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    APDBKeyLo_EL1 = X[t, 64];
        elsif PSTATE.EL == EL3 then
            APDBKeyLo_EL1 = 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.