RCWMASK_EL1, Read Check Write Instruction Mask (EL1)

The RCWMASK_EL1 characteristics are:

Purpose

Contains the mask used by RCW instructions.

Configuration

This register is present only when FEAT_THE is implemented. Otherwise, direct accesses to RCWMASK_EL1 are undefined.

RCWMASK_EL1 is a 128-bit register that can also be accessed as a 64-bit value. If it is accessed as a 64-bit register, accesses read and write bits [63:0] and do not modify bits [127:64].

Attributes

RCWMASK_EL1 is a:

- 128-bit register when FEAT_D128 is implemented
- 64-bit register otherwise

Field descriptions

When FEAT_D128 is implemented:

127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
Mask																															
95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
Mask																															
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
Mask																															
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
Mask																															

Mask, bits [127:0]

Mask used to decide which bit-fields are writable to the 128-bit Descriptor by RCW or RCWS Instructions.

The Effective value of Mask bit RCWMASK_EL1[n] is the same as RCWMASK_EL1[n], except as follows

- if n >= 17, and n <= 55, the Effective value of RCWMASK_EL1[n] is the same as RCWMASK_EL1[16].
- if n is in {126:125, 120:119, 114, 107:101, 90:56, 1:0}, the Effective value of RCWSMASK_EL1[n] is 0.

RCWMASK_EL1 register bits {126:125, 120:119, 114, 107:101, 90:17 1:0} are res0.

The reset behavior of this field is:

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

Otherwise:

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
Mask																															

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
Mask																															

Mask, bits [63:0]

Mask used to decide which bit-fields are writable to the 64-bit Descriptor by RCW or RCWS Instructions.

The Effective value of Mask bit RCWMASK_EL1[n] is the same as RCWMASK_EL1[n], except as follows

- if n >= 18, and n <= 49, the Effective value of RCWMASK_EL1[n] is the same as RCWMASK_EL1[17].

RCWMASK_EL1 register bits {52, 49:18, 0} are res0.

The reset behavior of this field is:

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

Accessing RCWMASK_EL1

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

```
MRS <Xt>, RCWMASK_EL1
```

op0	op1	CRn	CRm	op2

0b11	0b000	0b1101	0b0000	0b110

```
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.RCWMASKEn == '0' then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGRTR_EL2.nRCWMASK_EL1 ==
'0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = RCWMASK_EL1<63:0>;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.RCWMASKEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = RCWMASK_EL1<63:0>;
elsif PSTATE.EL == EL3 then
    X[t, 64] = RCWMASK_EL1<63:0>;
```

MSR RCWMASK_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1101	0b0000	0b110

```
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.RCWMASKEn == '0' then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGWTR_EL2.nRCWMASK_EL1 ==
'0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        RCWMASK_EL1<63:0> = 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.RCWMASKEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        RCWMASK_EL1<63:0> = X[t, 64];
elsif PSTATE.EL == EL3 then
    RCWMASK_EL1<63:0> = X[t, 64];
```

When FEAT_D128 is implemented

MRRS <Xt+1>, <Xt>, RCWMASK_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1101	0b0000	0b110

```
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.RCWMASKEn == '0' then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && SCR_EL3.D128En == '0'
then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGRTR_EL2.nRCWMASK_EL1 ==
'0' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif EL2Enabled() && (!IsHCRXEL2Enabled() ||
HCRX_EL2.D128En == '0') then
```

```
            AArch64.SystemAccessTrap(EL2, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    else
        (X[t + 1, 64], X[t, 64]) =
(RCWMASK_EL1<127:64>, RCWMASK_EL1<63:0>);
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.RCWMASKEn == '0' then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && SCR_EL3.D128En == '0'
then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    else
        (X[t + 1, 64], X[t, 64]) =
(RCWMASK_EL1<127:64>, RCWMASK_EL1<63:0>);
elsif PSTATE.EL == EL3 then
    (X[t + 1, 64], X[t, 64]) = (RCWMASK_EL1<127:64>,
RCWMASK_EL1<63:0>);
```

When FEAT_D128 is implemented

MSRR RCWMASK_EL1, <Xt+1>, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1101	0b0000	0b110

```
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.RCWMASKEn == '0' then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && SCR_EL3.D128En == '0'
then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGWTR_EL2.nRCWMASK_EL1 ==
'0' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif EL2Enabled() && (!IsHCRXEL2Enabled() ||
HCRX_EL2.D128En == '0') then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    else
        (RCWMASK_EL1<127:64>, RCWMASK_EL1<63:0>) =
(X[t + 1, 64], 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.RCWMASKEn == '0' then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && SCR_EL3.D128En == '0'
then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.RCWMASKEn == '0'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    else
        (RCWMASK_EL1<127:64>, RCWMASK_EL1<63:0>) =
(X[t + 1, 64], X[t, 64]);
elsif PSTATE.EL == EL3 then
    (RCWMASK_EL1<127:64>, RCWMASK_EL1<63:0>) = (X[t
+ 1, 64], X[t, 64]);
```