MSR (immediate)

Move immediate value to Special Register moves an immediate value to selected bits of the PSTATE. For more information, see *Process state, PSTATE*.

The bits that can be written by this instruction are:

- PSTATE.D, PSTATE.A, PSTATE.I, PSTATE.F, and PSTATE.SP.
- If *FEAT_SSBS* is implemented, PSTATE.SSBS.
- If *FEAT_PAN* is implemented, PSTATE.PAN.
- If *FEAT_UAO* is implemented, PSTATE.UAO.
- If *FEAT_DIT* is implemented, PSTATE.DIT.
- If *FEAT_MTE* is implemented, PSTATE.TCO.
- If *FEAT_NMI* is implemented, PSTATE.ALLINT.
- If *FEAT_SME* is implemented, PSTATE.SM and PSTATE.ZA.
- If FEAT_EBEP is implemented, PSTATE.PM.

This instruction is used by the aliases SMSTART, and SMSTOP.

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
1	1	0	1	0	1	0	1	0	0	0	0	0	op1			0	1	0	0	CRm				op2			1	1	1	1	1

MSR **<pstatefield>**, **#<imm>**

```
if op1 == '000' && op2 == '000' then SEE "CFINV";
if op1 == '000' && op2 == '001' then SEE "XAFLAG";
if op1 == '000' && op2 == '010' then SEE "AXFLAG";

AArch64.CheckSystemAccess('00', op1, '0100', CRm, op2, '11111', '0');
bits(2) min_EL;
boolean need_secure = FALSE;

case op1 of
    when '00x'
        min_EL = EL1;
    when '010'
        min_EL = EL1;
    when '011'
        min_EL = EL0;
    when '100'
        min_EL = EL2;
    when '101'
        if !IsFeatureImplemented(FEAT_VHE) then
            UNDEFINED;
        min_EL = EL2;
    when '110'
        min_EL = EL3;
    when '111'
        min_EL = EL1;
        need_secure = TRUE;

if (UInt(PSTATE.EL) < UInt(min_EL) || (need_secure && CurrentSecuritySt
    UNDEFINED;
```

```
PSTATEField field;
case op1:op2 of
    when '000 011'
        if !IsFeatureImplemented(FEAT_UAO) then UNDEFINED;
        field = PSTATEField_UAO;
    when '000 100'
        if !IsFeatureImplemented(FEAT_PAN) then UNDEFINED;
        field = PSTATEField_PAN;
    when '000 101' field = PSTATEField_SP;
    when '001 000'
        case CRm of
            when '000x'
                if !IsFeatureImplemented(FEAT_NMI) then UNDEFINED;
                field = PSTATEField_ALLINT;
            when '001x'
                if !IsFeatureImplemented(FEAT_EBEP) then UNDEFINED;
                field = PSTATEField_PM;
            otherwise
                UNDEFINED;
    when '011 010'
        if !IsFeatureImplemented(FEAT_DIT) then UNDEFINED;
        field = PSTATEField_DIT;
    when '011 011'
        case CRm of
            when '001x'
                if !IsFeatureImplemented(FEAT_SME) then UNDEFINED;
                field = PSTATEField_SVCRSM;
            when '010x'
                if !IsFeatureImplemented(FEAT_SME) then UNDEFINED;
                field = PSTATEField_SVCRZA;
            when '011x'
                if !IsFeatureImplemented(FEAT_SME) then UNDEFINED;
                field = PSTATEField_SVCRSMZA;
            otherwise
                UNDEFINED;
    when '011 100'
        if !IsFeatureImplemented(FEAT_MTE) then UNDEFINED;
        field = PSTATEField_TCO;
    when '011 110' field = PSTATEField_DAIFSet;
    when '011 111' field = PSTATEField_DAIFClr;
    when '011 001'
        if !IsFeatureImplemented(FEAT_SSBS) then UNDEFINED;
        field = PSTATEField_SSBS;
    otherwise UNDEFINED;
```

Assembler Symbols

<pstatefield>

Is a PSTATE field name. For the MSR instruction, this is encoded in “op1:op2:CRm”:

op1	op2	CRm	<pstatefield>	Architectural Feature
000	00x	xxxx	SEE PSTATE	-
000	010	xxxx	SEE PSTATE	-
000	011	xxxx	UAO	FEAT_UAO
000	100	xxxx	PAN	FEAT_PAN
000	101	xxxx	SPSel	-
000	11x	xxxx	RESERVED	-
001	000	000x	ALLINT	FEAT_NMI
001	000	001x	PM	FEAT_EBEP
001	000	01xx	RESERVED	-
001	000	1xxx	RESERVED	-
001	001	xxxx	RESERVED	-
001	01x	xxxx	RESERVED	-
001	1xx	xxxx	RESERVED	-
010	xxx	xxxx	RESERVED	-
011	000	xxxx	RESERVED	-
011	001	xxxx	SSBS	FEAT_SSBS
011	010	xxxx	DIT	FEAT_DIT
011	011	000x	RESERVED	-
011	011	001x	SVCRSM	FEAT_SME
011	011	010x	SVCRZA	FEAT_SME
011	011	011x	SVCRSMZA	FEAT_SME
011	011	1xxx	RESERVED	-
011	100	xxxx	TCO	FEAT_MTE
011	101	xxxx	RESERVED	-
011	110	xxxx	DAIFSet	-
011	111	xxxx	DAIFClr	-
1xx	xxx	xxxx	RESERVED	-

For the SMSTART and SMSTOP aliases, this is encoded in "CRm<2:1>", where 0b01 specifies SVCRSM, 0b10 specifies SVCRZA, and 0b11 specifies SVCRSMZA.

<imm>

Is a 4-bit unsigned immediate, in the range 0 to 15, encoded in the "CRm" field. Restricted to the range 0 to 1, encoded in "CRm<0>", when <pstatefield> is ALLINT, PM, SVCRSM, SVCRSMZA, or SVCRZA.

Alias Conditions

Alias	Is preferred when
SMSTART	`op1 == '011' && CRm == '0xx1' && op2 == '011'`
SMSTOP	`op1 == '011' && CRm == '0xx0' && op2 == '011'`

Operation

```
case field of
    when PSTATEField_SSBS
        PSTATE.SSBS = CRm<0>;
    when PSTATEField_SP
        PSTATE.SP = CRm<0>;
    when PSTATEField_DAIFSet
        AArch64.CheckDAIFAccess(PSTATEField_DAIFSet);
        PSTATE.D = PSTATE.D OR CRm<3>;
        PSTATE.A = PSTATE.A OR CRm<2>;
        PSTATE.I = PSTATE.I OR CRm<1>;
        PSTATE.F = PSTATE.F OR CRm<0>;
    when PSTATEField_DAIFClr
        AArch64.CheckDAIFAccess(PSTATEField_DAIFClr);
        PSTATE.D = PSTATE.D AND NOT(CRm<3>);
        PSTATE.A = PSTATE.A AND NOT(CRm<2>);
        PSTATE.I = PSTATE.I AND NOT(CRm<1>);
        PSTATE.F = PSTATE.F AND NOT(CRm<0>);
    when PSTATEField_PAN
        PSTATE.PAN = CRm<0>;
    when PSTATEField_UAO
        PSTATE.UAO = CRm<0>;
    when PSTATEField_DIT
        PSTATE.DIT = CRm<0>;
    when PSTATEField_TCO
        PSTATE.TCO = CRm<0>;
    when PSTATEField_ALLINT
        if (PSTATE.EL == EL1 && IsHCRXEL2Enabled() && HCRX_EL2.TALLINT
            AArch64.SystemAccessTrap(EL2, 0x18);
        PSTATE.ALLINT = CRm<0>;
    when PSTATEField_SVCRSM
        CheckSMEAccess();
        SetPSTATE_SM(CRm<0>);
    when PSTATEField_SVCRZA
        CheckSMEAccess();
        SetPSTATE_ZA(CRm<0>);
    when PSTATEField_SVCRSMZA
        CheckSMEAccess();
        SetPSTATE_SM(CRm<0>);
        SetPSTATE_ZA(CRm<0>);
    when PSTATEField_PM
        PSTATE.PM = CRm<0>;
```