GCSCR_EL1, Guarded Control Stack Control (EL1)

The GCSCR_EL1 characteristics are:

Purpose

Controls the Guarded control stack at EL1.

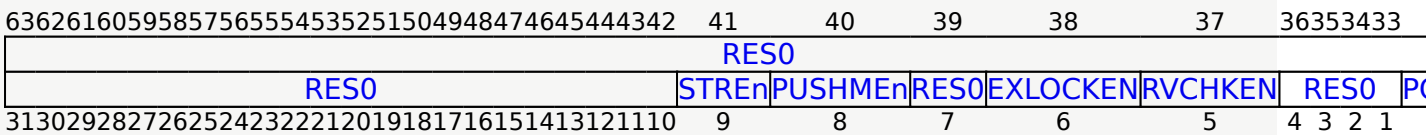
Configuration

This register is present only when FEAT_GCS is implemented. Otherwise, direct accesses to GCSCR_EL1 are undefined.

Attributes

GCSCR_EL1 is a 64-bit register.

Field descriptions



Bits [63:10]

Reserved, res0.

STREn, bit [9]

Execution of the following instructions are trapped:

- GCSSTR.
- GCSSTTR if any of the following are true.
  - PSTATE.UAO is 1.
  - If EL2 is implemented and enabled in the current Security state and HCR_EL2.{NV,NV1} is {1,1}.

STREn	Meaning
0b0	Execution of any of the specified instructions at EL1 cause a GCS exception.
0b1	This control does not cause any instructions to be trapped.

The reset behavior of this field is:

- On a Warm reset, this field resets to `0`.

PUSHMEn, bit [8]

Trap GCSPUSHM instruction.

PUSHMEn	Meaning
`0b0`	Execution of a GCSPUSHM instruction at EL1 causes a Trap exception.
`0b1`	This control does not cause any instructions to be trapped.

The reset behavior of this field is:

- On a Warm reset, this field resets to `0`.

Bit [7]

Reserved, res0.

EXLOCKEN, bit [6]

Exception state lock.

Prevents `MSR` instructions from writing to ELR_EL1 or SPSR_EL1.

EXLOCKEN	Meaning
`0b0`	EL1 exception state locking disabled.
`0b1`	EL1 exception state locking enabled.

The reset behavior of this field is:

- On a Warm reset, this field resets to `0`.

RVCHKEN, bit [5]

Return value check enable.

RVCHKEN	Meaning
`0b0`	Return value checking disabled at EL1.
`0b1`	Return value checking enabled at EL1.

The reset behavior of this field is:

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

Bits [4:1]

Reserved, res0.

PCRSEL, bit [0]

Guarded control stack procedure call return enable selection.

PCRSEL	Meaning
0b0	Guarded control stack at EL1 is not PCR Selected.
0b1	Guarded control stack at EL1 is PCR Selected.

The reset behavior of this field is:

- On a Warm reset, this field resets to 0.

Accessing GCSCR_EL1

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

MRS <Xt>, GCSCR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0101	0b000

```
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.GCSEn == '0' then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGRTR_EL2.nGCS_EL1 == '0'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.GCSEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
```

```
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
'111' then
        X[t, 64] = NVMem[0x8D0];
    else
        X[t, 64] = GCSCR_EL1;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.GCSEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.GCSEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HCR_EL2.E2H == '1' then
        X[t, 64] = GCSCR_EL2;
    else
        X[t, 64] = GCSCR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = GCSCR_EL1;
```

MSR GCSCR_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0101	0b000

```
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.GCSEn == '0' then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGWTR_EL2.nGCS_EL1 == '0'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.GCSEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
'111' then
        NVMem[0x8D0] = X[t, 64];
    else
        GCSCR_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.GCSEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.GCSEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HCR_EL2.E2H == '1' then
        GCSCR_EL2 = X[t, 64];
    else
        GCSCR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    GCSCR_EL1 = X[t, 64];
```

MRS <Xt>, GCSCR_EL12

op0	op1	CRn	CRm	op2
0b11	0b101	0b0010	0b0101	0b000

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '101'
then
        X[t, 64] = NVMem[0x8D0];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        if Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && SCR_EL3.GCSEn == '0'
then
            UNDEFINED;
        elsif HaveEL(EL3) && SCR_EL3.GCSEn == '0'
then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = GCSCR_EL1;
    else
        UNDEFINED;
elsif PSTATE.EL == EL3 then
    if EL2Enabled() && !ELUsingAArch32(EL2) &&
HCR_EL2.E2H == '1' then
        X[t, 64] = GCSCR_EL1;
    else
        UNDEFINED;
```

MSR GCSCR_EL12, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b101	0b0010	0b0101	0b000

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '101'
then
        NVMem[0x8D0] = X[t, 64];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        if Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && SCR_EL3.GCSEn == '0'
then
            UNDEFINED;
        elsif HaveEL(EL3) && SCR_EL3.GCSEn == '0'
then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            GCSCR_EL1 = X[t, 64];
    else
        UNDEFINED;
elsif PSTATE.EL == EL3 then
    if EL2Enabled() && !ELUsingAArch32(EL2) &&
HCR_EL2.E2H == '1' then
        GCSCR_EL1 = X[t, 64];
    else
        UNDEFINED;
```


28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94