TRCIDR2, ID Register 2

The TRCIDR2 characteristics are:

Purpose

Returns the tracing capabilities of the trace unit.

Configuration

AArch64 System register TRCIDR2 bits [31:0] are architecturally mapped to External register TRCIDR2[31:0].

This register is present only when FEAT_ETE is implemented and FEAT_TRC_SR is implemented. Otherwise, direct accesses to TRCIDR2 are undefined.

Attributes

TRCIDR2 is a 64-bit register.

Field descriptions

<table>
<tr><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td></tr>
<tr><td colspan="32">RES0</td></tr>
<tr><td>WFXMODE</td><td colspan="2">VMIDOPT</td><td colspan="4">CCSIZE</td><td colspan="5">DVSIZE</td><td colspan="5">DASIZE</td><td colspan="5">VMIDSIZE</td><td colspan="5">CIDSIZE</td><td colspan="5">IASIZE</td></tr>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Bits [63:32]

Reserved, res0.

WFXMODE, bit [31]

Indicates whether `WFI`, `WFIT`, `WFE`, and `WFET` instructions are classified as P0 instructions:

WFXMODE	Meaning
`0b0`	`WFI`, `WFIT`, `WFE`, and `WFET` instructions are not classified as P0 instructions.
`0b1`	`WFI`, `WFIT`, `WFE`, and `WFET` instructions are classified as P0 instructions.

VMIDOPT, bits [30:29]

Indicates the options for Virtual context identifier selection.

VMIDOPT	Meaning
`0b00`	Virtual context identifier selection not supported. TRCCONFIGR.VMIDOPT is res0.
`0b01`	Virtual context identifier selection supported. TRCCONFIGR.VMIDOPT is implemented.
`0b10`	Virtual context identifier selection not supported. TRCCONFIGR.VMIDOPT is res1.

All other values are reserved.

If TRCIDR2.VMIDSIZE == `0b00000` then this field is `0b00`.

If TRCIDR2.VMIDSIZE != `0b00000` then this field is `0b10`.

CCSIZE, bits [28:25]
When TRCIDR0.TRCCCI == 1:

Indicates the size of the cycle counter.

CCSIZE	Meaning
`0b0000`	The cycle counter is 12 bits in length.
`0b0001`	The cycle counter is 13 bits in length.
`0b0010`	The cycle counter is 14 bits in length.
`0b0011`	The cycle counter is 15 bits in length.
`0b0100`	The cycle counter is 16 bits in length.
`0b0101`	The cycle counter is 17 bits in length.
`0b0110`	The cycle counter is 18 bits in length.
`0b0111`	The cycle counter is 19 bits in length.
`0b1000`	The cycle counter is 20 bits in length.

All other values are reserved.

Otherwise:

Reserved, res0.

DVSIZE, bits [24:20]

When TRCIDR0.TRCDATA != 0b00:

Indicates the data value size in bytes. Data tracing is not implemented in ETE and this field is reserved for other trace architectures. Allocated in other trace architectures.

DVSIZE	Meaning
`0b00000`	Data value tracing not implemented.
`0b00100`	Data value tracing has a maximum of 32-bit data values.
`0b01000`	Data value tracing has a maximum of 64-bit data values.

All other values are reserved.

Otherwise:

Reserved, res0.

DASIZE, bits [19:15]

When TRCIDR0.TRCDATA != 0b00:

Indicates the data address size in bytes. Data tracing is not implemented in ETE and this field is reserved for other trace architectures. Allocated in other trace architectures.

DASIZE	Meaning
`0b00000`	Data address tracing not implemented.
`0b00100`	Data address tracing has a maximum of 32-bit data addresses.
`0b01000`	Data address tracing has a maximum of 64-bit data addresses.

All other values are reserved.

Otherwise:

Reserved, res0.

VMIDSIZE, bits [14:10]

Indicates the trace unit Virtual context identifier size.

VMIDSIZE	Meaning
`0b00000`	Virtual context identifier tracing is not supported.
`0b00001`	8-bit Virtual context identifier size.
`0b00010`	16-bit Virtual context identifier size.
`0b00100`	32-bit Virtual context identifier size.

All other values are reserved.

If the PE does not implement EL2 then this field is `0b00000`.

If the PE implements EL2 then this field is `0b00100`.

CIDSIZE, bits [9:5]

Indicates the Context identifier size.

CIDSIZE	Meaning
`0b00000`	Context identifier tracing is not supported.
`0b00100`	32-bit Context identifier size.

All other values are reserved.

This field reads as `0b00100`.

IASIZE, bits [4:0]

Virtual instruction address size.

IASIZE	Meaning
`0b00100`	Maximum of 32-bit instruction address size.
`0b01000`	Maximum of 64-bit instruction address size.

All other values are reserved.

This field reads as `0b01000`.

Accessing TRCIDR2

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

```
MRS <Xt>, TRCIDR2
```

op0	op1	CRn	CRm	op2
0b10	0b001	0b0000	0b1010	0b111

```
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'" && CPTR_EL3.TTA == '1' then
        UNDEFINED;
    elsif CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.TRCID == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR2;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && CPTR_EL3.TTA == '1' then
        UNDEFINED;
    elsif CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR2;
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR2;
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94