SYS S1_<op1>_<Cn>_<Cm>_<op2>, SYSL S1_<op1>_<Cn>_<Cm>_<op2>, SYSP S1_<op1>_<Cn>_<Cm>_<op2>, IMPLEMENTATION DEFINED maintenance instructions

The SYS S1_<op1>_<Cn>_<Cm>_<op2>, SYSL S1_<op1>_<Cn>_<Cm>_<op2>, SYSP S1_<op1>_<Cn>_<Cm>_<op2> characteristics are:

Purpose

This area of the System instruction encoding space is reserved for implementation defined System instructions.

Configuration

There are no configuration notes.

Attributes

SYS S1_<op1>_<Cn>_<Cm>_<op2>, SYSL S1_<op1>_<Cn>_<Cm>_<op2>, SYSP S1_<op1>_<Cn>_<Cm>_<op2> is a:

- 128-bit System instruction when FEAT_SYSINSTR128 is implemented
- 64-bit System instruction otherwise

Field descriptions

When FEAT_SYSINSTR128 is implemented:

<table>
<tr><td>127</td><td>126</td><td>125</td><td>124</td><td>123</td><td>122</td><td>121</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td><td>112</td><td>111</td><td>110</td><td>109</td><td>108</td><td>107</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td><td>96</td></tr>
<tr><td colspan="32">IMPLEMENTATION DEFINED</td></tr>
<tr><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td><td>80</td><td>79</td><td>78</td><td>77</td><td>76</td><td>75</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td><td>64</td></tr>
<tr><td colspan="32">IMPLEMENTATION DEFINED</td></tr>
<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">IMPLEMENTATION DEFINED</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>
<tr><td colspan="32">IMPLEMENTATION DEFINED</td></tr>
</table>

IMPLEMENTATION DEFINED, bits [127:0]

implementation defined.

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	
IMPLEMENTATION DEFINED	
IMPLEMENTATION DEFINED	
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	

IMPLEMENTATION DEFINED, bits [63:0]

implementation defined.

Executing SYS S1_<op1>_<Cn>_<Cm>_<op2>, SYSL S1_<op1>_<Cn>_<Cm>_<op2>, SYSP S1_<op1>_<Cn>_<Cm>_<op2>

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

```
SYS #<op1>, <Cn>, <Cm>, #<op2>{, <Xt>}
```

op0	op1	CRn	CRm	op2
0b01	op1[2:0]	0b1x11	Cm[3:0]	op2[2:0]

```
if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11') && SCTLR_EL1.TIDCP == '1' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11' && SCTLR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.ImpDefSysInstr(1, op1, CRn, CRm, op2, t);
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.ImpDefSysInstr(1, op1, CRn, CRm, op2, t);
elsif PSTATE.EL == EL2 then
    AArch64.ImpDefSysInstr(1, op1, CRn, CRm, op2, t);
elsif PSTATE.EL == EL3 then
    AArch64.ImpDefSysInstr(1, op1, CRn, CRm, op2, t);
```

SYSL <Xt>, #<op1>, <Cn>, <Cm>, #<op2>

op0	op1	CRn	CRm	op2
0b01	op1[2:0]	0b1x11	Cm[3:0]	op2[2:0]

```
if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11')
&& SCTLR_EL1.TIDCP == '1' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
&& SCTLR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.ImpDefSysInstrWithResult(1, op1,
CRn, CRm, op2);
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.ImpDefSysInstrWithResult(1, op1,
CRn, CRm, op2);
elsif PSTATE.EL == EL2 then
    AArch64.ImpDefSysInstrWithResult(1, op1, CRn,
CRm, op2);
elsif PSTATE.EL == EL3 then
    AArch64.ImpDefSysInstrWithResult(1, op1, CRn,
CRm, op2);
```

When FEAT_SYSINSTR128 is implemented

SYSP #<op1>, <Cn>, <Cm>, #<op2>{, <Xt>, <Xt2>}

op0	op1	CRn	CRm	op2
0b01	op1[2:0]	0b1x11	Cm[3:0]	op2[2:0]

```
if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11')
&& SCTLR_EL1.TIDCP == '1' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x14);
        else
            AArch64.SystemAccessTrap(EL1, 0x14);
    elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
&& SCTLR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    else
        AArch64.ImpDefSysInstr128(1, op1, CRn, CRm,
op2, t, t2);
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    else
        AArch64.ImpDefSysInstr128(1, op1, CRn, CRm,
op2, t, t2);
elsif PSTATE.EL == EL2 then
    AArch64.ImpDefSysInstr128(1, op1, CRn, CRm, op2,
t, t2);
elsif PSTATE.EL == EL3 then
    AArch64.ImpDefSysInstr128(1, op1, CRn, CRm, op2,
t, t2);
```

AArch32 Registers | AArch64 Registers | AArch32 Instructions | AArch64 Instructions | Index by Encoding | External Registers

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94