RCWSWPP, RCWSWPPA, RCWSWPPL, RCWSWPPAL

Read Check Write Swap quadword in memory atomically loads a 128-bit quadword from a memory location, and conditionally stores the value held in a pair of registers back to the same memory location. Storing back to memory is conditional on RCW Checks. The value initially loaded from memory is returned in the same pair of registers. This instruction updates the condition flags based on the result of the update of memory.

- RCWSWPPA and RCWSWPPAL load from memory with acquire semantics.
- RCWSWPPL and RCWSWPPAL store to memory with release semantics.
- RCWSWPP has neither acquire nor release semantics.

Integer
(FEAT_D128 && FEAT_THE)

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
0	0	0	1	1	0	0	1	A	R	1	Rt2	1	0	1	0	0	0	Rn	Rt
	S												o3		opc				

RCWSWPP (A == 0 && R == 0)

        RCWSWPP <Xt1>, <Xt2>, [<Xn|SP>]

RCWSWPPA (A == 1 && R == 0)

        RCWSWPPA <Xt1>, <Xt2>, [<Xn|SP>]

RCWSWPPAL (A == 1 && R == 1)

        RCWSWPPAL <Xt1>, <Xt2>, [<Xn|SP>]

RCWSWPPL (A == 0 && R == 1)

        RCWSWPPL <Xt1>, <Xt2>, [<Xn|SP>]

```
if !IsFeatureImplemented(FEAT_D128) || !IsFeatureImplemented(FEAT_THE)
if Rt == '11111' then UNDEFINED;
if Rt2 == '11111' then UNDEFINED;
integer t = UInt(Rt);
integer t2 = UInt(Rt2);
integer n = UInt(Rn);

boolean acquire = A == '1';
boolean release = R == '1';
boolean tagchecked = n != 31;
```

```
    boolean rt_unknown = FALSE;

    if t == t2 then
        Constraint c = ConstrainUnpredictable(Unpredictable_LSE128OVERLAP);
        assert c IN {Constraint_UNKNOWN, Constraint_UNDEF, Constraint_NOP};
        case c of
            when Constraint_UNKNOWN rt_unknown = TRUE;     // result is UNKN
            when Constraint_UNDEF    UNDEFINED;
            when Constraint_NOP      EndOfInstruction();
```

Assembler Symbols

<Xt1>   Is the 64-bit name of the first general-purpose register to be
              transferred, encoded in the "Rt" field.

<Xt2>   Is the 64-bit name of the second general-purpose register to
              be transferred, encoded in the "Rt2" field.

<Xn|SP>   Is the 64-bit name of the general-purpose base register or
              stack pointer, encoded in the "Rn" field.

Operation

```
    if !IsD128Enabled(PSTATE.EL) then UNDEFINED;
    bits(64) address;
    bits(64) value1;
    bits(64) value2;
    bits(128) newdata;
    bits(128) readdata;
    bits(4) nzcv;

    AccessDescriptor accdesc = CreateAccDescRCW(MemAtomicOp_SWP, FALSE, acc

    if n == 31 then
        CheckSPAlignment();
        address = SP[];
    else
        address = X[n, 64];

    value1 = X[t, 64];
    value2 = X[t2, 64];

    newdata = if BigEndian(accdesc.acctype) then value1:value2 else value2:

    bits(128) compdata = bits(128) UNKNOWN;    // Irrelevant when not execu
    (nzcv, readdata) = MemAtomicRCW(address, compdata, newdata, accdesc);

    PSTATE.<N,Z,C,V> = nzcv;
    if rt_unknown then
        readdata = bits(128) UNKNOWN;

    if BigEndian(accdesc.acctype) then
        X[t, 64] = readdata<127:64>;
        X[t2, 64] = readdata<63:0>;
    else
        X[t, 64] = readdata<63:0>;
        X[t2, 64] = readdata<127:64>;
```

Operational information

If PSTATE.DIT is 1, the timing of this instruction is insensitive to the value of the data being loaded or stored.

Internal version only: isa v33.64, AdvSIMD v29.12, pseudocode no_diffs_2023_09_RC2, sve v2023-06_rel ; Build timestamp: 2023-09-18T17:56