HINT

Hint instruction is for the instruction set space that is reserved for architectural hint instructions.
Some encodings described here are not allocated in this revision of the architecture, and behave as NOPs. These encodings might be allocated to other hint functionality in future revisions of the architecture and therefore must not be used by software.

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

HINT #<imm>

```
SystemHintOp op;

case CRm:op2 of
    when '0000 000' op = SystemHintOp_NOP;
    when '0000 001' op = SystemHintOp_YIELD;
    when '0000 010' op = SystemHintOp_WFE;
    when '0000 011' op = SystemHintOp_WFI;
    when '0000 100' op = SystemHintOp_SEV;
    when '0000 101' op = SystemHintOp_SEVL;
    when '0000 110'
        if !IsFeatureImplemented(FEAT_DGH) then EndOfInstruction();
        op = SystemHintOp_DGH;
    when '0000 111' SEE "XPACLRI";
    when '0001 xxx'
        case op2 of
            when '000' SEE "PACIA1716";
            when '010' SEE "PACIB1716";
            when '100' SEE "AUTIA1716";
            when '110' SEE "AUTIB1716";
            otherwise EndOfInstruction();
    when '0010 000'
        if !IsFeatureImplemented(FEAT_RAS) then EndOfInstruction();
        op = SystemHintOp_ESB;
    when '0010 001'
        if !IsFeatureImplemented(FEAT_SPE) then EndOfInstruction();
        op = SystemHintOp_PSB;
    when '0010 010'
        if !IsFeatureImplemented(FEAT_TRF) then EndOfInstruction();
        op = SystemHintOp_TSB;
    when '0010 011'
        if !IsFeatureImplemented(FEAT_GCS) then EndOfInstruction();
        op = SystemHintOp_GCSB;
    when '0010 100'
        op = SystemHintOp_CSDB;
    when '0010 110'
        if !IsFeatureImplemented(FEAT_CLRBHB) then
            EndOfInstruction();
        op = SystemHintOp_CLRBHB;
    when '0011 xxx'
        case op2 of
```

```
            when '000' SEE "PACIAZ";
            when '001' SEE "PACIASP";
            when '010' SEE "PACIBZ";
            when '011' SEE "PACIBSP";
            when '100' SEE "AUTIAZ";
            when '101' SEE "AUTIASP";
            when '110' SEE "AUTIBZ";
            when '111' SEE "AUTIBSP";
    when '0100 xx0'
        op = SystemHintOp_BTI;
        // Check branch target compatibility between BTI instruction an
        SetBTypeCompatible(BTypeCompatible_BTI(op2<2:1>));
    when '0101 000'
        if !IsFeatureImplemented(FEAT_CHK) then EndOfInstruction();
        op = SystemHintOp_CHKFEAT;
    otherwise EndOfInstruction();
```

Assembler Symbols

<imm> Is a 7-bit unsigned immediate, in the range 0 to 127, encoded in the "CRm:op2" field.

The encodings that are allocated to architectural hint functionality are described in the 'Hints' table in the 'Index by Encoding'.

Note

For allocated encodings of "CRm:op2":

- A disassembler will disassemble the allocated instruction, rather than the `HINT` instruction.
- An assembler may support assembly of allocated encodings using `HINT` with the corresponding <imm> value, but it is not required to do so.

Operation

```
case op of
    when SystemHintOp_YIELD
        Hint_Yield();

    when SystemHintOp_DGH
        Hint_DGH();

    when SystemHintOp_WFE
        integer localtimeout = 1 << 64;     // No local timeout event is
        Hint_WFE(localtimeout, WFxType_WFE);

    when SystemHintOp_WFI
        integer localtimeout = 1 << 64;     // No local timeout event is
        Hint_WFI(localtimeout, WFxType_WFI);
```

```
    when SystemHintOp_SEV
        SendEvent();

    when SystemHintOp_SEVL
        SendEventLocal();

    when SystemHintOp_ESB
        if IsFeatureImplemented(FEAT_TME) && TSTATE.depth > 0 then
            FailTransaction(TMFailure_ERR, FALSE);
        SynchronizeErrors();
        AArch64.ESBOperation();
        if PSTATE.EL IN {EL0, EL1} && EL2Enabled() then AArch64.vESBOpe
        TakeUnmaskedSErrorInterrupts();

    when SystemHintOp_PSB
        ProfilingSynchronizationBarrier();

    when SystemHintOp_TSB
        TraceSynchronizationBarrier();

    when SystemHintOp_GCSB
        GCSSynchronizationBarrier();

    when SystemHintOp_CHKFEAT
        X[16, 64] = AArch64.ChkFeat(X[16, 64]);

    when SystemHintOp_CSDB
        ConsumptionOfSpeculativeDataBarrier();

    when SystemHintOp_CLRBHB
        Hint_CLRBHB();

    when SystemHintOp_BTI
        SetBTypeNext('00');

    when SystemHintOp_NOP
        return;    // do nothing

    otherwise
        Unreachable();
```

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