SM4E

SM4 encryption and decryption

The SM4E instruction reads 16 bytes of input data from each 128-bit segment of the first source vector, together with four iterations of 32-bit round keys from the corresponding 128-bit segments of the second source vector. Each block of data is encrypted by four rounds in accordance with the SM4 standard, and destructively placed in the corresponding segments of the first source vector. This instruction is unpredicated.

ID_AA64ZFR0_EL1.SM4 indicates whether this instruction is implemented.

This instruction is illegal when executed in Streaming SVE mode, unless FEAT_SME_FA64 is implemented and enabled.

SVE2
(FEAT_SVE_SM4)

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	1	0	0	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	Zm					Zdn				
								size<1>	size<0>																						

SM4E <Zdn>.S, <Zdn>.S, <Zm>.S

```
if !HaveSVE() || !HaveSVE2SM4() then UNDEFINED;
integer m = UInt(Zm);
integer dn = UInt(Zdn);
```

Assembler Symbols

<Zdn> Is the name of the first source and destination scalable vector register, encoded in the "Zdn" field.

<Zm> Is the name of the second source scalable vector register, encoded in the "Zm" field.

Operation

```
CheckNonStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer segments = VL DIV 128;
bits(VL) operand1 = Z[dn, VL];
bits(VL) operand2 = Z[m, VL];
bits(VL) result;

for s = 0 to segments-1
    bits(128) key = Elem[operand2, s, 128];
    bits(32) intval;
    bits(8) sboxout;
    bits(128) roundresult = Elem[operand1, s, 128];
    bits(32) roundkey;
```

```
        for index = 0 to 3
            roundkey = Elem[key, index, 32];
            intval = roundresult<127:96> EOR roundresult<95:64> EOR roundre

            for i = 0 to 3
                Elem[intval, i,8]  = Sbox(Elem[intval,i,8]);

            intval = intval EOR ROL(intval, 2) EOR ROL(intval,10) EOR ROL(i
            intval = intval EOR roundresult<31:0>;

            roundresult<31:0> = roundresult<63:32>;
            roundresult<63:32> = roundresult<95:64>;
            roundresult<95:64> = roundresult<127:96>;
            roundresult<127:96> = intval;

        Elem[result, s, 128] = roundresult;

    Z[dn, VL] = result;
```

Operational information

If PSTATE.DIT is 1:

- The execution time of this instruction is independent of:
  - The values of the data supplied in any of its registers.
  - The values of the NZCV flags.
- The response of this instruction to asynchronous exceptions does not vary based on:
  - The values of the data supplied in any of its registers.
  - The values of the NZCV flags.

Base Instructions SIMD&FP Instructions SVE Instructions SME Instructions Index by Encoding Sh Pseu

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