REV

Reverse Bytes reverses the byte order in a register.

This instruction is used by the pseudo-instruction [REV64](#).

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
sf	1	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	x			Rn					Rd		

opc

32-bit (sf == 0 && opc == 10)

```
        REV <Wd>, <Wn>
```

64-bit (sf == 1 && opc == 11)

```
        REV <Xd>, <Xn>
```

```
    if opc == '11' && sf == '0' then UNDEFINED;

    integer d = UInt(Rd);
    integer n = UInt(Rn);

    constant integer datasize = 32 << UInt(sf);
    constant integer container_size = 8 << UInt(opc);
```

Assembler Symbols

<Wd>   Is the 32-bit name of the general-purpose destination register, encoded in the "Rd" field.

<Wn>   Is the 32-bit name of the general-purpose source register, encoded in the "Rn" field.

<Xd>   Is the 64-bit name of the general-purpose destination register, encoded in the "Rd" field.

<Xn>   Is the 64-bit name of the general-purpose source register, encoded in the "Rn" field.

Operation

```
    bits(datasize) operand = X[n, datasize];
    bits(datasize) result;

    constant integer containers = datasize DIV container_size;
    for c = 0 to containers-1
        bits(container_size) container = Elem[operand, c, container_size];
        Elem[result, c, container_size] = Reverse(container, 8);

    X[d, datasize] = 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.