MOVI

Move Immediate (vector). This instruction places an immediate constant into every vector element of the destination SIMD&FP register.

Depending on the settings in the *CPACR_EL1*, *CPTR_EL2*, and *CPTR_EL3* registers, and the current Security state and Exception level, an attempt to execute the instruction might be trapped.

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	Q	op	0	1	1	1	1	0	0	0	0	0	a	b	c	cmode	0	1	d	e	f	g	h	Rd

8-bit (op == 0 && cmode == 1110)

        MOVI <Vd>.<T>, #<imm8>{, LSL #0}

16-bit shifted immediate (op == 0 && cmode == 10x0)

        MOVI <Vd>.<T>, #<imm8>{, LSL #<amount>}

32-bit shifted immediate (op == 0 && cmode == 0xx0)

        MOVI <Vd>.<T>, #<imm8>{, LSL #<amount>}

32-bit shifting ones (op == 0 && cmode == 110x)

        MOVI <Vd>.<T>, #<imm8>, MSL #<amount>

64-bit scalar (Q == 0 && op == 1 && cmode == 1110)

        MOVI <Dd>, #<imm>

64-bit vector (Q == 1 && op == 1 && cmode == 1110)

        MOVI <Vd>.2D, #<imm>

```
integer rd = UInt(Rd);

constant integer datasize = 64 << UInt(Q);
bits(datasize) imm;
bits(64) imm64;

ImmediateOp operation;
case cmode:op of
    when '0xx00' operation = ImmediateOp_MOVI;
    when '0xx01' operation = ImmediateOp_MVNI;
    when '0xx10' operation = ImmediateOp_ORR;
    when '0xx11' operation = ImmediateOp_BIC;
```

```
        when '10x00' operation = ImmediateOp_MOVI;
        when '10x01' operation = ImmediateOp_MVNI;
        when '10x10' operation = ImmediateOp_ORR;
        when '10x11' operation = ImmediateOp_BIC;
        when '110x0' operation = ImmediateOp_MOVI;
        when '110x1' operation = ImmediateOp_MVNI;
        when '1110x' operation = ImmediateOp_MOVI;
        when '11110' operation = ImmediateOp_MOVI;
        when '11111'
            // FMOV Dn,#imm is in main FP instruction set
            if Q == '0' then UNDEFINED;
            operation = ImmediateOp_MOVI;

    imm64 = AdvSIMDExpandImm(op, cmode, a:b:c:d:e:f:g:h);
    imm = Replicate(imm64, datasize DIV 64);
```

Assembler Symbols

<Dd>
Is the 64-bit name of the SIMD&FP destination register, encoded in the "Rd" field.

<Vd>
Is the name of the SIMD&FP destination register, encoded in the "Rd" field.

<imm>
Is a 64-bit immediate 'aaaaaaaabbbbbbbbccccccccddddddddeeeeeeeeffffffffggggggggghhhhhhh encoded in "a:b:c:d:e:f:g:h".

<T>

For the 8-bit variant: is an arrangement specifier, encoded in "Q":

Q	<T>
0	8B
1	16B

For the 16-bit variant: is an arrangement specifier, encoded in "Q":

Q	<T>
0	4H
1	8H

For the 32-bit variant: is an arrangement specifier, encoded in "Q":

Q	<T>
0	2S
1	4S

<imm8>
Is an 8-bit immediate encoded in "a:b:c:d:e:f:g:h".

For the 16-bit shifted immediate variant: is the shift amount encoded in "cmode<1>":

cmode<1>	<amount>
0	0
1	8

defaulting to 0 if LSL is omitted.

For the 32-bit shifted immediate variant: is the shift amount encoded in "cmode<2:1>":

cmode<2:1>	<amount>
00	0
01	8
10	16
11	24

defaulting to 0 if LSL is omitted.

For the 32-bit shifting ones variant: is the shift amount encoded in "cmode<0>":

cmode<0>	<amount>
0	8
1	16

Operation

```
CheckFPAdvSIMDEnabled64();
bits(datasize) operand;
bits(datasize) result;

case operation of
    when ImmediateOp_MOVI
        result = imm;
    when ImmediateOp_MVNI
        result = NOT(imm);
    when ImmediateOp_ORR
        operand = V[rd, datasize];
        result = operand OR imm;
    when ImmediateOp_BIC
        operand = V[rd, datasize];
        result = operand AND NOT(imm);

V[rd, 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.