ADD (immediate)

Add (immediate) adds a register value and an optionally-shifted immediate value, and writes the result to the destination register.

This instruction is used by the alias [MOV (to/from SP)](#).

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	0	0	1	0	0	0	1	0	sh		imm12	Rn	Rd

op S

32-bit (sf == 0)

```
        ADD <Wd|WSP>, <Wn|WSP>, #<imm>{, <shift>}
```

64-bit (sf == 1)

```
        ADD <Xd|SP>, <Xn|SP>, #<imm>{, <shift>}
```

```
integer d = UInt(Rd);
integer n = UInt(Rn);
constant integer datasize = 32 << UInt(sf);
bits(datasize) imm;

case sh of
    when '0' imm = ZeroExtend(imm12, datasize);
    when '1' imm = ZeroExtend(imm12:Zeros(12), datasize);
```

Assembler Symbols

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

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

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

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

<imm>   Is an unsigned immediate, in the range 0 to 4095, encoded in the "imm12" field.

<shift>

Is the optional left shift to apply to the immediate, defaulting to LSL #0 and encoded in "sh":

sh	<shift>
0	LSL #0
1	LSL #12

Alias Conditions

Alias	Is preferred when
MOV (to/ from SP)	sh == '0' && imm12 == '000000000000' && (Rd == '11111' || Rn == '

Operation

```
bits(datasize) result;
bits(datasize) operand1 = if n == 31 then SP[]<datasize-1:0> else X[n,

(result, -) = AddWithCarry(operand1, imm, '0');

if d == 31 then
    SP[] = ZeroExtend(result, 64);
else
    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.