`SP` See *Register names* on page C1-145.

`Wn` See *Register names* on page C1-145.

`WSP` See *Register names* on page C1-145.

`WZR` See *Register names* on page C1-145.

`Xn` See *Register names* on page C1-145.

`XZR` See *Register names* on page C1-145

### C1.2.3 Instruction Mnemonics

The A64 assembly language overloads instruction mnemonics and distinguishes between the different forms of an instruction based on the operand types. For example, the following `ADD` instructions all have different opcodes. However, the programmer must only remember one mnemonic, as the assembler automatically chooses the correct opcode based on the operands. The disassembler follows the same procedure in reverse.

**Example C1-1 ADD instructions with different opcodes**

```
ADD W0, W1, W2              // add 32-bit register
ADD X0, X1, X2              // add 64-bit register
ADD X0, X1, W2, SXTW        // add 64-bit extended register
ADD X0, X1, #42             // add 64-bit immediate
```

### C1.2.4 Condition code

The A64 ISA has some instructions that set Condition flags or test Condition codes or both. For information about instructions that set the Condition flags or use the condition mnemonics, see *Condition flags and related instructions* on page C6-525.

Table C1-1 shows the available Condition codes.

**Table C1-1 Condition codes**

| cond | Mnemonic | Meaning (integer) | Meaning (floating-point)[a] | Condition flags |
|---|---|---|---|---|
| 0000 | EQ | Equal | Equal | Z == 1 |
| 0001 | NE | Not equal | Not equal or unordered | Z == 0 |
| 0010 | CS or HS | Carry set | Greater than, equal, or unordered | C == 1 |
| 0011 | CC or LO | Carry clear | Less than | C == 0 |
| 0100 | MI | Minus, negative | Less than | N == 1 |
| 0101 | PL | Plus, positive or zero | Greater than, equal, or unordered | N == 0 |
| 0110 | VS | Overflow | Unordered | V == 1 |
| 0111 | VC | No overflow | Ordered | V == 0 |
| 1000 | HI | Unsigned higher | Greater than, or unordered | C ==1 && Z == 0 |
| 1001 | LS | Unsigned lower or same | Less than or equal | !(C ==1 && Z ==0) |
| 1010 | GE | Signed greater than or equal | Greater than or equal | N == V |
| 1011 | LT | Signed less than | Less than, or unordered | N! = V |

**Table C1-1 Condition codes (continued)**

| cond | Mnemonic | Meaning (integer) | Meaning (floating-point)[a] | Condition flags |
|---|---|---|---|---|
| 1100 | GT | Signed greater than | Greater than | Z == 0 && N == V |
| 1101 | LE | Signed less than or equal | Less than, equal, or unordered | !(Z == 0 && N == V) |
| 1110 | AL | Always | Always | Any |
| 1111 | NV[b] | Always | Always | Any |

a. Unordered means at least one NaN operand.

b. The Condition code NV exists only to provide a valid disassembly of the 0b1111 encoding, otherwise its behavior is identical to AL.

## C1.2.5 Register names

This section describes the AArch64 registers. It contains the following subsections:

- *General-purpose register file and zero register and stack pointer*.
- *SIMD and floating-point register file* on page C1-146.
- *SIMD and floating-point scalar register names* on page C1-146.
- *SIMD vector register names* on page C1-147.
- *SIMD vector element names* on page C1-147.

### General-purpose register file and zero register and stack pointer

The 31 general-purpose registers in the general-purpose register file are named R0-R30 and encoded in the instruction register fields with values 0-30. In a general-purpose register field the value 31 represents either the current stack pointer or the zero register, depending on the instruction and the operand position.

When the registers are used in a specific instruction variant, they must be qualified to indicate the operand data size, 32 bits or 64 bits, and the data size of the instruction.

When the data size is 32 bits, the lower 32 bits of the register are used and the upper 32 bits are ignored on a read and cleared to zero on a write.

Table C1-2 shows the qualified names for registers, where *n* is a register number 0-30.

**Table C1-2 Naming of general-purpose registers, the zero register, and the stack pointer**

| Name | Size | Encoding | Description |
|---|---|---|---|
| Wn | 32 bits | 0-30 | General-purpose register 0-30 |
| Xn | 64 bits | 0-30 | General-purpose register 0-30 |
| WZR | 32 bits | 31 | Zero register |
| XZR | 64 bits | 31 | Zero register |
| WSP | 32 bits | 31 | Current stack pointer |
| SP | 64 bits | 31 | Current stack pointer |

This list gives more information about the instruction arguments shown in Table C1-2:

- The names Xn and Wn both refer to the same general-purpose register, Rn.
- There is no register named W31 or X31.