"VLDR S0,=1.0" rejected, but accepts "VLDR S0,=1" and generates bad code.

Ref: https://bugs.launchpad.net/gcc-arm-embedded/+bug/1695572

The instruction VLDR S0,=1.0 is rejected by the assembler, but VLDR S0,=1 is accepted. However, using an integer constant does NOT load the floating-point 1.0 (0x3F800000) into S0; instead it loads the integer representation of 1 (i.e., 0x00000001) into S0.

(Has NOT been corrected as of version 12.2.0-1 of GNU ARM Embedded Toolchain.)

gnu assembler can't deal with AL condition code in IT block

Ref: https://bugs.launchpad.net/gcc-arm-embedded/+bug/1620025

As a result, the coding technique used in section 6.5.1 of the 4th and 5th editions of the text to reduce code size currently will not assemble.

(Has NOT been corrected as of version 12.2.0-1 of GNU ARM Embedded Toolchain.)

VLDR/VSTR S0,[R0,R1] accepted; assembles as VLDR/VSTR S0,[R0]

Ref: https://bugs.launchpad.net/gcc-arm-embedded/+bug/1996779

The assembler accepts without any warning or error message VSTR and VLDR instructions with the memory addressing mode of [Rn, Rm]. This memory addressing mode does not exist for these instructions, as the modes are limited to [Rn] and [Rn, #imm]. But the assembler accepts them anyway, and then produces code that ignores the Rm index register.

(Has NOT been corrected as of version 12.2.0-1 of GNU ARM Embedded Toolchain.)