3.9.2 Floating Point Numbers

Floating point numbers are stored in one of two IEEE 754 standard formats that uses adjacent 16-bit words: 32-bit single precision or 64-bit double precision.

The REAL data type represents single precision floating point numbers. The LREAL data type represents double precision floating point numbers. REAL and LREAL variables are typically used to store data from analog I/O devices, calculated values, and constants.

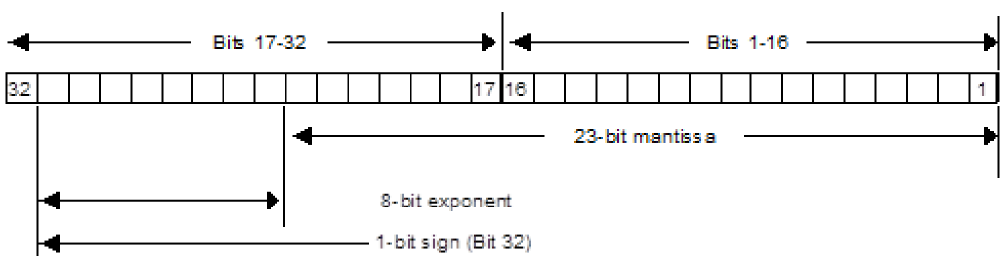
Types of Floating-Point Variables

Data Type	Precision and Range
REAL	Limited to 6 or 7 significant digits, with a range of approximately ±1.401298x10^{-45} through ±3.402823x10^{38}.
LREAL	Limited to 17 significant digits, with a range of approximately ±2.2250738585072020x10^{-308} to ±1.7976931348623157x10^{308}.

Note: *The programming software allows 32-bit and 64-bit arguments (DWORD, DINT, REAL, and LREAL) to be placed in discrete memories such as %I, %M, and %R in the PACSystems target. This is not allowed on Series 90-70 targets. (Note that any bit reference address that is passed to a non-bit parameter must be byte-aligned. This is the same as the Series 90-70 CPU.)*

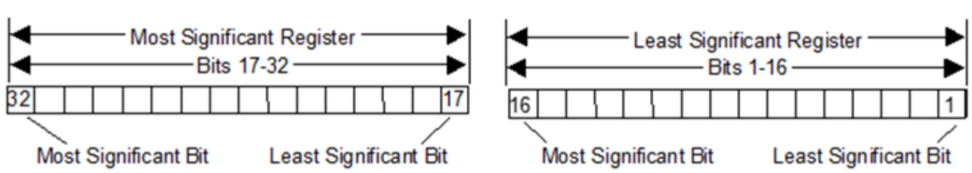
Internal Format of REAL Numbers

Figure 17



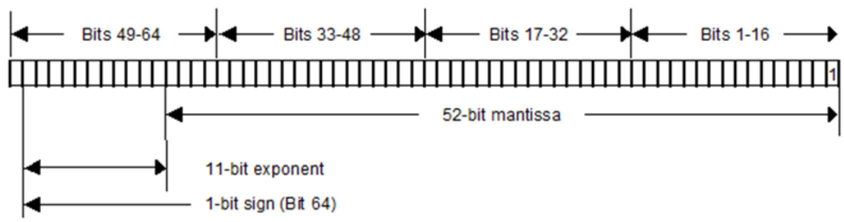
Register use by a single floating-point number is diagrammed below. For example, if the floating-point number occupies registers R5 and R6, R5 is the least significant register and R6 is the most significant register.

Figure 18



Internal Format of LREAL Numbers

Figure 19



Errors in Floating Point Numbers and Operations

Overflow occurs when a REAL or LREAL function generates a number outside the allowed range. When this occurs, the Enable Out output of the function is set Off, and the result is set to positive infinity (for a number greater than the upper limit) or negative infinity (for a number less than the lower limit). You can determine where this occurs by testing the sense of the Enable Out output.

Binary representations of Infinity and NaN values have exponents that contain all 1s.

IEEE 754 Infinity Representations

	REAL	LREAL
POS_INF (positive infinity)	= 7F800000h	= 7FF0000000000000h
NEG_INF (negative infinity)	= FF800000h	= 7FF0000000000001h

If the infinities produced by overflow are used as operands to other REAL or LREAL functions, they may cause an undefined result. This undefined result is referred to as an NaN (Not a Number). For example, the result of adding positive infinity to negative infinity is undefined. When the ADD_REAL function is invoked with positive infinity and negative infinity as its operands, it produces an NaN. If any operand of a function is a NaN, the result will be some NaN.

Note: *For NaN, the Enable Out output is Off (not energized).*

IEEE 754 Representations of NaN values:

REAL	LREAL
7F800001 through 7FFFFFFF	7FF8000000000001 through 7FFFFFFFFFFFFFFF
FF800001 through FFFFFFFF	FFF0000000000001 through FFFFFFFFFFFFFFFF

Note: *For releases 5.0 and greater, the CPU may return slightly different values for NaN compared to previous releases. In some cases, the result is a special type of NaN displayed as #IND in Machine Edition. In these cases, for example, EXP(-infinity), power flow out of the function is identical to that in previous releases.*