- Sign bit: 1 bit
- Exponent: 11 bits
- Significand precision: 53 bits (52 explicitly stored)

The sign bit determines the sign of the number (including when this number is zero, which is signed).

The exponent field can be interpreted as either an 11-bit signed integer from −1024 to 1023 (2's complement) or an 11-bit unsigned integer from 0 to 2047, which is the accepted biased form in the IEEE 754 binary64 definition. If the unsigned integer format is used, the exponent value used in the arithmetic is the exponent shifted by a bias – for the IEEE 754 binary64 case, an exponent value of 1023 represents the actual zero (i.e. for 2^{e-1023} to be one, e must be 1023). Exponents range from −1022 to +1023 because exponents of −1023 (all 0s) and +1024 (all 1s) are reserved for special numbers.

The 53-bit significand precision gives from 15 to 17 significant decimal digits precision ($2^{-53} \approx 1.11 \times 10^{-16}$). If a decimal string with at most 15 significant digits is converted to IEEE 754 double-precision representation, and then converted back to a decimal string with the same number of digits, the final result should match the original string. If an IEEE 754 double-precision number is converted to a decimal string with at least 17 significant digits, and then converted back to double-precision representation, the final result must match the original number.[1]