Floating-Point Numbers - Special Values*

Charles Severance
Kevin Dowd

In addition to specifying the results of operations on numeric data, the IEEE standard also specifies the precise behavior on undefined operations such as dividing by zero. These results are indicated using several special values. These values are bit patterns that are stored in variables that are checked before operations are performed. The IEEE operations are all defined on these special values in addition to the normal numeric values. Table 1: Special Values for an IEEE 32-Bit Number summarizes the special values for a 32-bit IEEE floating-point number.

Special Values for an IEEE 32-Bit Number

Special Value	Exponent	Significand
+ or − 0	00000000	0
Denormalized number	00000000	nonzero
NaN (Not a Number)	11111111	nonzero
+ or − Infinity	11111111	0

Table 1

The value of the exponent and significand determines which type of special value this particular floating-point number represents. Zero is designed such that integer zero and floating-point zero are the same bit pattern.

Denormalized numbers can occur at some point as a number continues to get smaller, and the exponent has reached the minimum value. We could declare that minimum to be the smallest representable value. However, with denormalized values, we can continue by setting the exponent bits to zero and shifting the significand bits to the right, first adding the leading "1" that was dropped, then continuing to add leading zeros to indicate even smaller values. At some point the last nonzero digit is shifted off to the right, and the value becomes zero. This approach is called *gradual underflow* where the value keeps approaching zero and then eventually becomes zero. Not all implementations support denormalized numbers in hardware; they might trap to a software routine to handle these numbers at a significant performance cost.

At the top end of the biased exponent value, an exponent of all 1s can represent the *Not a Number* (NaN) value or infinity. Infinity occurs in computations roughly according to the principles of mathematics.

*Version 1.3: Aug 25, 2010 10:38 am -0500
†http://creativecommons.org/licenses/by/3.0/

If you continue to increase the magnitude of a number beyond the range of the floating-point format, once the range has been exceeded, the value becomes infinity. Once a value is infinity, further additions won't increase it, and subtractions won't decrease it. You can also produce the value infinity by dividing a nonzero value by zero. If you divide a nonzero value by infinity, you get zero as a result.

The NaN value indicates a number that is not mathematically defined. You can generate a NaN by dividing zero by zero, dividing infinity by infinity, or taking the square root of -1. The difference between infinity and NaN is that the NaN value has a nonzero significand. The NaN value is very sticky. Any operation that has a NaN as one of its inputs always produces a NaN result.