The Character of Binary8 Floating-Point Formats

Jeffrey Sarnoff

2023-Oct-10

This presentation is appropriate for the six formats: binary8p2 .. binary8p7. The remaining binary8 formats (binary8p0, binary8p1) are not amenable to this characterization.

p is the precision of the format in bits (p includes the implicit bit of the significand)

Constants

All binary8 formats have 2^k == 2^8 == 256 unique values and distinct encodings. Values are either Normal, Subnormal or one of {0, +Inf, -Inf, NaN}. Identifying the values that are neither normal nor subnormal as “Special”, there are 4 Special Values. Identifying all values except the special values as “Ordinary”, there are 256 – 4 = 252 Ordinary Values. Precisely half of the ordinary values are positively signed and half are negatively signed. Magnitudes are unsigned ordinary values, as there are equal number of positive and negative ordinary values, we may take the positive ordinary values as magnitudes.

```
k                 is the number of bits required to encode any binary8
n_values          is the number of unique encodings in a binary8 value set
n_specials        tallys all the non[sub]normal values {0, +Inf, -Inf, NaN}
n_ordinaries      tallys all the normal and subnormal values
```

```
k = 8
n_values        = 2^k                               # 256
n_specials      = 4
n_ordinaries    = n_values - n_specials             # 252
n_ordinary_mags = n_ordinaries >> 1                 # 126
```

Parameters

```
w(p)              is the  bitwidth of exponent field             derived from k, p
t(p)              is the explicit significand bit count          derived from p
emax(p)           is the maximum value of unbiased exponent      defined from p, w(p)
bias(p)           is the exponent bias, emax(p) + 1              defined from p, w(p)
```

```
w(p) = 8 - p                                        # k - p
t(p) = p – 1                                        # one less than precision
emax(p) = 2^(7 - p) - 1                             # bias(p) - 1, 2^(w(p) - 1) - 1
bias(p) = 2^(7 - p)                                 # emax(p) + 1, 2^(w(p) - 1)
```

Counts

Counting Significands

There are as many unique explicit significands (ignoring the implicit bit) as unique normal significands (significands with the implicit bit set to 0b1). There are 2 fewer subnormal significands than there are normal significands: 0x00 and 0x080 (Zero and NaN) are neither normal nor subnormal values; and each exponent bitfield is zeroed (i.e. they would have been subnormals).

The number of unique significands for normal magnitudes:

```
2^(precision - 1) == 2^(explicit significand bits)
```

There are twice that number of signed significands for normal values:

```
2^(precision) == 2*(2^(explicit significand bits)
```

Since those counts cover both positive and negative values, the significand count for normal and subnormal magnitudes is half of the normal [subnormal] significands. There are 2 fewer subnormal signed significands than the normal signed count. There is 1 less subnormal magnitude significand than the normal count.

```
n_normal_significands(p)       = 2^(p)
n_normal_significand_mags(p)   = 2^(p - 1)

n_subnormal_significands(p)    = n_normal_significands(p) - 2      # 2^(p) - 2
n_subnormal_significand_mags(p) = n_normal_significand_mags(p) - 1   # 2^(p-1) - 1
```

Counting Values

(a) Subnormal values are interpreted with an implicit leading 0 bit.
(b) All subnormal values have a raw exponent field that is filled with zero bits.
(c) Moreover, any value with a zeroed raw exponent field is a subnormal value.

```
n_subnormals(p)     = n_subnormal_significands(p)
n_subnormal_mags(p) = n_subnormal_significand_mags(p)
```

The number of normal values is the total value count less the subnormal and special value counts. This is equal to the the number of "ordinary" values less the number of subnormal values. There are half as many normal magnitudes as there are signed normal values.

```
n_normals(p)     = n_ordinaries - n_subnormals(p)
n_normal_mags(p) = n_normals(p) >> 1

n_normal_significands(p)        = 2^(p - 1)
n_normal_significand_mags(p)    = n_normal_significands(p) >> 1

n_subnormal_significands(p)     = n_normal_significands(p) - 2
n_subnormal_significand_mags(p) = n_subnormal_significands(p) >> 1
```

Extremal Magnitudes

(a) The least nonzero magnitude is the subnormal with raw exponent == 0b0 and significand == 0b1.
(b) The least normal magnitude is the normal with raw exponent == 0b1 and significand == 0b0.
(c) The greatest subnormal magnitude has raw exponent == 0b0 and an explicit significand of 1 bits.
(d) The greatest normal magnitude is given with raw exponent of 1 bits and significand of 0b1..10.

```
recip(x) = 1 / x                                    # rewrite to use rationals

max_normal_mag(p)     = recip( (2^emax(p)))         # 1 / (2^7 – p) – 1))
min_subnormal_mag(p)  = max_normal_mag(p) * recip( n_subnormal_mags(p) )

max_subnormal_mag(p)  = max_normal_mag(p) - min_subnormal_mag(p)
max_normal_mag(p)     = (n_subnormals(p) * recip( 2^(p - 2) )) * (2^emax(p))
```