Sensonics Devices

A POSIT TUTORIAL

SR5.1: POSIT SUMMARY

$$(-1)^s \times (2^{2^{es}})^k \times 2^e \times \left(1 + \sum_{i=1}^{N-2-es-m} f_i \times 2^{-i}\right)$$

- Posit's tapered accuracy has a key consequence: there is no "sub-normal" number representation.

- The fractional portion of Posit (equivalent to IEEE 754's "mantissa") *always* has the implied 1 → complexity of "de-normal" numbers in IEEE 754 gone! There is only one rounding mode.

- There is only 1 representation for zero in Posit. An all-zero sequence of N bits is zero. Zero also has only one sign → positive and negative zero (as in IEEE 754) is gone!

- ± infinity is represented by a single code: 1 followed by N-1 zeroes.

- It is expected that exceptions that would otherwise have resulted in NaN should generate hardware "Traps" that can then be processed in Software.

SR5.1: POSIT SUMMARY - 2

There is a very interesting property of POSITs: The regime, exponent and fraction bits are 2's complemented if the sign bit is 1, but left untouched otherwise.

There is another feature in Posit called the "Quire" function. The idea is to compute the following computations with a single rounding at the end of the total computation (not after each operation):

$$x=(a+b)\times c$$
$$x=a\times b-(c\times d)$$
$$x=\Sigma\, ai$$
$$x=\Sigma\, aibi$$

We have implemented a 512 bit Quire Register.

Our preliminary estimates indicate that SR5.1 with a POSIT Compute Engine will operate at 700+MHz at the 28 nm node

SR5.1: POSIT SUMMARY - 3

It is to be noted that **es** is a representation system parameter and cannot be inferred from the number. However, Gustafson[1] suggests the following formula:

$$es = log_2 N\text{-}3$$

POSIT representation has a fixed (1) bit reserved for sign. The remaining bits are allocated respectively to "regime", "e", and "fraction". **e** can be between zero bits and **es** bits, e.g., with es = 3, we can have 0, 1, 2, or 3 bits for **e**. When 1 or 2 bits are used for **e,** these are the more significant bits. For example, if es is 2 and there is a 1 bit exponent = 1, it is deemed to be 10'b = 2 → $2^e = 2^{10'b} = 2^2 = 4$ and not $2^e = 2^{01'b} = 2^1 = 2$.

A way to interpret **es** is the exponent (of 2) that determines the range around ±1 where full precision is desired. (e.g., if es is 2, this implies full precision for $-8 \le x \le -1$ and $1 \le x \le 8$)

This accuracy tapers downwards to zero in both the direction towards 0 and towards ± ∞

[1]See Page 42 of "POSIT Arithmetic" by John Gustafson, Oct 10, 2017.

SR5.1: POSIT SUMMARY - 4

At the (low precision) limits a number is represented in Posit as follows:

sign	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	...	r_{N-1}

In this category when:

Sign is 0 and all regime bits are also 0 this is zero
Sign is 1 and all regime bits are 0, this is ± ∞
Sign is 0 and N-2 regime bits are 0 followed by a 0→ 1 transition → +ve; k = -m = -(N-2)
Sign is 1 and N-2 regime bits are 0 followed by a 0→ 1 transition, → –ve; k = -m = -(N-2)
Sign is 0 and N-1 regime bits are 1 with an <u>implicit</u> 1→ 0 transition, → k = (m-1) = N-2
Sign is 1 and N-1 regime bits are 1 (→ an <u>implicit</u> 1→ 0 transition) → k = (m-1) = N-2

SR5.1: POSIT SUMMARY - 5

The smallest number positive number in Posit (N, es) is ***minpos***:

$$2^{-(N-2)\times 2^{es}}$$

The largest positive number called ***maxpos*** in Posit (N, es) is:

$$2^{(N-2)\times 2^{es}}$$

At the moderate precision ranges, where regime bits are traded off with the exponent bits the representation is as follows:

sign	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	...	r_{N-3}	t	e_0

sign	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	...	r_{N-2-es}	t	e_0	...	e_{es-1}

When regime makes way for 1 exponent bit, k = ±(N-2-1) and the Posit is then:

$$\pm 2^{10\pm(N-3)\times 2^{es}}$$

(10 above is binary → 2; for es = 3, this would have been deemed to be 100 → 4)

SR5.1: POSIT SUMMARY - 6

At the full precision ranges, regime bits are traded off with the fraction bits as follows:

sign	r_1	r_2	r_3	r_4	r_5	r_6	r_7	...	r_{N-3-es}	t	e_0	...	e_{es-1}	f_1

sign	r_1	r_2	r_3	r_4	...	r_m	t	e_0	...	e_{es-1}	f_1	f_2	...	$f_{(N-3-es-m)}$

sign	r_1	t	e_0	...	e_{es-1}	f_1	f_2	f_3	f_4	f_5	f_6	f_7	...	$f_{(N-3-es)}$

The tradeoff between regime (m) and fraction bits varies with k going all the way down to 0 and the fraction going all the way up to N-2-es bits. When k is zero it apparent that the number is in full accuracy and in the range:

$$\pm 2^{e} \times \left(1 + \sum_{i=1}^{N-2-es} f_i 2^{-i}\right)$$

SR5.1: POSIT ACCURACY

- POSIT (32,2) can have at best a guaranteed absolute accuracy of 8.4 digit ($2^{-28} \sim 10^{-8.4}$)
- We ensure 8 - 9 digit accuracy of result for every math function being computed in POSIT
- The Quire mechanism ensures that chain operations as described earlier do not exhibit accuracy fall-off due to rounding even for fairly deep chains

→ ***POSIT has the potential to consistently deliver > 8 digits of final accuracy***
→ ***It is generally believed (!) that this is the achieved <u>end performance</u> of Double Precision Floating Point***

Thank you