A Note on Subsecond Base-2 Time Intervals

N. E. Davis `~lagrev-nocfep`
Urbit Foundation

Abstract

Urbit time provides for the representation of subsecond values down to 2^{-64} s. Such a base-two numbering system prompts the consideration of prefixes for subsecond intervals that are analogous to the SI prefixes for positive powers of two, as used in memory sizes. We propose a set of prefixes for subsecond intervals that are similar to the SI prefixes for negative powers of ten, and we suggest that these prefixes be used in Urbit time representations. We also discuss the relative error in these prefixes compared to the SI prefixes for negative powers of ten.

Urbit time is conventionally a 128-bit atom,[1] with the lower 64 bits denoting fractions of a second and the upper 64 bits denoting multiples of a second. (That is, 1 s = `(bex 64)`.)

By expanding the atomic form of time (`@dr`) to the tuple form, we can clearly see how each component of the time is represented:

```
> (yell now)  :: Treating "now" as an interval.
[d=106.751.991.823.894 h=15 m=19 s=22 f=~[0xd43]]
```

[1]Time can actually be an arbitrarily sized atom, but 2^{64} s $\approx 58 \times 10^9$ a; most discussions of noncosmological time will not involve such large intervals.

Manuscript submitted for review.

Address author correspondence to `~lagrev-nocfep`.

The last component of the tuple is a list of subsecond time intervals; while `now` does not feign to provide more accuracy than 2^{-16} s, the underlying representation can provide for subsecond intervals down to 2^{-64} s by including more values in the `f` list.

```
> (yell `@dr`0x1)
[d=0 h=0 m=0 s=0 f=~[0x0 0x0 0x0 0x1]]
```

Subsecond decimal time does not cleanly map to this representation, and the resulting hexadecimal values are largely opaque to human interpretation:

Interval	**Decimal Value**	**Hexadecimal Value**
1 ms	10^{-3} s	`0x41.8937.4bc6.a7ef`
1 µs	10^{-6} s	`0x10c6.f7a0.b5ed`
1 ns	10^{-9} s	`0x4.4b82.fa09`
1 ps	10^{-12} s	`0x119.7998`
1 fs	10^{-15} s	`0x480e`
1 as	10^{-18} s	`0x12`

Smaller intervals of a second than 2^{-64} cannot be represented in native Urbit time relative time `@dr`.

Urbit time can therefore be more exactly expressed in terms of binary fractions of a second rather than decimal fractions of a second. However, while there are established expressions for powers of two that are close to positive powers of ten, there are no expressions for powers of two close to negative powers of ten. We would like to introduce such a scheme for convenience in Urbit time.

Binary Value	**Hexadecimal Value**
2^{-10} s	`0x40.0000.0000.0000`
2^{-20} s	`0x10.0000.0000.0000`
2^{-30} s	`0x4.0000.0000`
2^{-40} s	`0x100.0000`
2^{-50} s	`0x4000`
2^{-60} s	`0x10`

What prefixes should we provide? The standard names for the positive values of ten are:

Binary Value	Prefix	Approximation
2^{10}	kibi	10^{3}
2^{20}	mebi	10^{6}
2^{30}	gibi	10^{9}
2^{40}	tebi	10^{12}
2^{50}	pebi	10^{15}
2^{60}	exbi	10^{18}

This suggests that we should prefer prefixes that are similar to the standard names for the negative values of two but with a different last syllable. Following a suggestion by `~rovyns-ricfer` for "-ki-", we propose the following prefixes:

Binary Value	Prefix	Symbol	Approximation
2^{-10}	miki	mi	10^{-3}
2^{-20}	muki	µi or ui	10^{-6}
2^{-30}	naki	ni	10^{-9}
2^{-40}	piki	pi	10^{-12}
2^{-50}	feki	fi	10^{-15}
2^{-60}	akki	ai	10^{-18}

As occurs with memory values denominated in kibi, mebi, gibi, etc., it will be important to keep in mind the relative error in such time expressions. We can compare these binary magnitudes roughly to decimal magnitudes and calculate the resulting error, much as we do with the SI prefixes for positive values of ten and two:

Binary Value	Prefix	Approximation	Error
2^{-10}	miki	10^{-3}	2.4%
2^{-20}	muki	10^{-6}	4.9%
2^{-30}	naki	10^{-9}	7.4%
2^{-40}	piki	10^{-12}	9.9%
2^{-50}	feki	10^{-15}	12.6%
2^{-60}	akki	10^{-18}	12.5%[2]

We trust that the current ability to express subsecond intervals in Urbit time will continue to prove useful, and furthermore that the proposed prefixes will promote legibility and accuracy.

[2] This is due to truncation error; it "should" be 15%.