PLL has a jitter accumulation effect; jitter introduced during a cycle is fed back through the ring oscillator on the next cycle. The jitter accumulation is inversely proportional to the PLL loop bandwidth. If you expand the loop bandwidth, though, you allow in high-frequency components of jitter on the reference input.

DLL does not “recirculate” its jitter and, therefore, has no accumulation, but the DLL passes whatever noise is present on the reference clock directly in.

PLLs better when noise on the reference clock dominates. DLL has much simpler loop dynamics.
DLL linear analysis

\[
D_o(s) = (D_I(s) - D_o(s)) f_{ref} \frac{I_{ch}}{sC} K_{DL}
\]

\[
\frac{D_o(s)}{D_i(s)} = \frac{1}{1 + s / \omega_n}
\]

\[
\omega_n = I_{ch} K_{DL} f_{ref} \frac{1}{C}
\]
Self-biased DLL
Digital DLL

- May get dither at lock
- Deadband can be introduced to control dither.

Dither is also a problem with “high-gain” phase detectors (bang-bang type control)
DLL issues

- Two main DLL disadvantages:
  - Clock jitter propagation
  - DLLs locking to multiples of the clock period (false lock)
  - Start-up problem with PFDs
  - Two-sided constraints on delay line… may be difficult to lock.
Start-up problem with PFD

A down pulse

Better off with JK-latched-based PD.
Semidigital dual DLL