PLL Optimisation for White Rabbit Applications

Pedro Moreira
University College London
19/10/2010
Agenda

- Digital Dual Time Difference Circuit
  - Helper PLL
    - Control Loop Design
    - Implementation
    - Results
  - Main PLL
    - Control Loop Design
    - Implementation
    - Results
- Frequency Multipliers
- Conclusions
White Rabbit Project

Which PLLs?
Digital Dual Mixer Time Difference

- **DDMTD Circuit**

\[ u_{1}(t) \]

\[ u_{2}(t) \]

\[ v_{dmtd} = \frac{N}{N+1} v_n \]

\[ \Delta t(t) = \Delta t_{beat}(t) \frac{v_{beat}}{v_n} \]

\[ \Delta t_{min} = \frac{1}{v_{dmtd} v_n} = \frac{v_n - v_{dmtd}}{v_{dmtd} v_n} = \frac{1}{Nv_n} \]
Helper PLL

- Control Loop

\[ v_{out} = \frac{N}{N + 1} v_{in} \]

\[ N = 2^{13} \]

\[ v_{pd} \approx 15 257 \text{ kHz} \]

- Design Goal
- Low jitter

\[ \Delta t = 960 \text{ fs} \]
**Helper PLL**

**Control Design**

\[
G(s) = \frac{\theta_o}{\theta_e} = \frac{K_d K_o F(s)}{s} \\
H(s) = \frac{\theta_o}{\theta_i} = \frac{G(s)}{1 + G(s)} = \frac{K_d K_o F(s)}{s + K_d K_o F(s)} \\
F(s) = K_1 + \frac{K_2}{s} \\
H(s) = \frac{K_d K_o (K_1 s + K_2)}{s^2 + s K_d K_o K_1 + K_d K_o K_2}
\]
Helper PLL

Control Design

\[ H(s) = \frac{K_d K_o (K_1 s + K_2)}{s^2 + s K_d K_o K_1 + K_d K_o K_2} \]

\[ H(s) = \frac{2\zeta \omega_n s + \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]

\[ \omega_n = \sqrt{K_d K_o K_2} \]

\[ \zeta = \frac{K_1}{2} \sqrt{\frac{K_d K_o}{K_2}} \]

Digital Domain

- \( F(z) = \text{c2d}(F(s), 1/(fs/N+1), 'tustin'); \) % bilinear approximation
- Fix-Point conversion (Scaling)
Helper PLL

- Phase Detector
- TDC + BB Phase detector

\[ K_d = \frac{v_{sys}}{v_{beat} \cdot 2 \cdot \pi} \approx 1.3 \times 10^3 \text{ticks/rad} \]
Helper PLL

- VXCO + PLL fanout Modelling

\[ y = 0.6240x + 124978697 \]

\[ K_0 = 2 \pi 0.6240 \approx 3.9207 \ \frac{\omega}{\text{bit}} \]
Helper PLL

- Implementation in Hardware

![Diagram of PLL implementation in hardware]
Helper PLL

- Loop Design

<table>
<thead>
<tr>
<th>Model</th>
<th>H_n</th>
<th>zeta</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop2</td>
<td>0.65</td>
<td>1.685</td>
</tr>
<tr>
<td>loop3</td>
<td>1.9</td>
<td>1.838</td>
</tr>
<tr>
<td>loop4</td>
<td>0.19</td>
<td>1.931</td>
</tr>
<tr>
<td>loop5</td>
<td>0.02</td>
<td>1.92</td>
</tr>
<tr>
<td>loop6</td>
<td>0.19</td>
<td>0.193</td>
</tr>
<tr>
<td>loop7</td>
<td>0.19</td>
<td>0.002</td>
</tr>
</tbody>
</table>
Helper PLL

- Implementation Results

<table>
<thead>
<tr>
<th>Model</th>
<th>RMS Jitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCXO</td>
<td>15.6 ps</td>
</tr>
<tr>
<td>Ref</td>
<td>23.0 ps</td>
</tr>
<tr>
<td>loop2</td>
<td>32.4 ps</td>
</tr>
<tr>
<td>loop3</td>
<td>55.8 ps</td>
</tr>
<tr>
<td>loop4</td>
<td>42.7 ps</td>
</tr>
<tr>
<td>loop5</td>
<td>18.3 ps</td>
</tr>
<tr>
<td>loop6</td>
<td>23.4 ps</td>
</tr>
<tr>
<td>loop7</td>
<td>17.2 ps</td>
</tr>
</tbody>
</table>

![Graph showing frequency response](image-url)
Main PLL

- Control Loop

- Design Goal
  Low jitter
Main PLL

- Phase Detector (DDMTD)

\[ N = 2^{13} \]
\[ v_{dmtd} = \frac{N}{N + 1} v_n = 124\,984\,743 \text{ MHz} \]
\[ v_{beat} = 15\,257 \text{ kHz} \]
\[ K_d = \frac{\text{sys clk} / v_{beat}}{2\pi} = 1.303 \times 10^3 \text{(ticks/rad)} \]
Main PLL

- VCTCXO Modelling + PLL Fanout

\[ y = 0.1037x + 124996240 \]

\[ K_0 = \frac{1}{2 \pi 0.1037 \omega} \approx 1.534 \frac{\text{bit}}{\text{Hz}} \]
Main PLL

- Loop Controller $F(s)$

$$H(s) = \frac{K_d K_o (K_1 s + K_2)}{s^2 + s K_d K_o K_1 + K_d K_o K_2}$$

$$H(s) = \frac{2 \zeta \omega_n s + \omega_n^2}{s^2 + 2 \zeta \omega_n s + \omega_n^2}$$

$$\omega_n = \sqrt{K_d K_o K_2}$$

$$\zeta = \frac{K_1}{2} \sqrt{\frac{K_d K_o}{K_2}}$$

- Digital Domain

  - $F(z) = \text{c2d}(F(s), 1/(fs/N+1), 'tustin');$ % bilinear approximation
  
  - Fix-Point conversion (Scaling)
Main PLL

Implementation in Hardware
Main PLL

- Results

![Bode Diagram]

<table>
<thead>
<tr>
<th>Model</th>
<th>H_n</th>
<th>zeta</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop2</td>
<td>12</td>
<td>1.49</td>
</tr>
<tr>
<td>loop3</td>
<td>53</td>
<td>0.26</td>
</tr>
<tr>
<td>loop7</td>
<td>92</td>
<td>0.26</td>
</tr>
<tr>
<td>loop8</td>
<td>131</td>
<td>0.26</td>
</tr>
<tr>
<td>loop9</td>
<td>186</td>
<td>0.62</td>
</tr>
<tr>
<td>loop10</td>
<td>787</td>
<td>0.26</td>
</tr>
</tbody>
</table>
Main PLL

- Results

<table>
<thead>
<tr>
<th>Model</th>
<th>RMS Jitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCTXCO</td>
<td>3.3 ps</td>
</tr>
<tr>
<td>Ref</td>
<td>12 ps</td>
</tr>
<tr>
<td>loop2</td>
<td>2.5 ps</td>
</tr>
<tr>
<td>loop3</td>
<td>3.0 ps</td>
</tr>
<tr>
<td>loop4</td>
<td>3.0 ps</td>
</tr>
<tr>
<td>loop7</td>
<td>4.5 ps</td>
</tr>
<tr>
<td>loop8</td>
<td>5.0 ps</td>
</tr>
<tr>
<td>loop9</td>
<td>7.5 ps</td>
</tr>
</tbody>
</table>
Frequency Multipliers

- CDCM61004 (Helper PLL)
  - Sampling Frequency - 25 MHz
  - Bandwidth - 400 KHz (internal)

- AD9516 (Main PLL)
  - PD sampling frequency - 25 MHz (it was 5 MHz)
  - Bandwidth - 300 KHz (external)
  - 10 MHz and 25 MHz references share the same bandwidth
Conclusions

- Design of the helper PLL
  - PD Noise floor is -40 dBc/Hz
- Design of the main PLL
  - DDMTD noise floor is -80 dBc/Hz
  - Optimal parameter results in 2.5 ps
- Frequency Multipliers (AD1916)
  - Increasing the PD sampling frequency reduces PLL noise floor
  - Different bandwidths for the 10 MHz and 25 MHz references
Q & A

Thank you for the Attention!
PLL Optimal Cutoff

Offset Frequency (Hz) vs. dBc/Hz graph showing:
- VCTXCO Jitter = 3.2871e-12
- Reference Jitter = 1.2741e-11
- PLL Jitter = 1.6470e-12

Graph indicates significant jitter suppression with optimal cutoff.
Cascading PLLs
PLL noise = 2.462e-12
Cascade 10 PLL noise = 3.8898e-12
Master CLK, Slave CLK
17 Sep 2010 – the case that a derived or recovered clock in Synchronous Ethernet ... This design was optimized for low jitter peaking, < 0.1 dB typical, and to ...
AD9516-