ECEN720: High-Speed Links
Circuits and Systems
Spring 2015

Lecture 12: CDRs

Sam Palermo
Analog & Mixed-Signal Center
Texas A&M University
Announcements

- Preliminary Project Report #1 due today
  - Project Overview
  - Literature Survey
  - Initial Architecture (it can change completely)
  - Initial Simulation Results
    - Must have at least one initial simulation result

- No class on Tuesday 4/21
Agenda

• CDR overview
• CDR phase detectors
• Dual-loop CDRs
• CDR circuits
  • Phase interpolators
  • Delay-locked loops
• CDR jitter properties
Embedded Clock I/O Circuits

- TX PLL
- TX Clock Distribution
- CDR
  - Per-channel PLL-based
  - Dual-loop w/ Global PLL &
    - Local DLL/PI
    - Local Phase-Rotator PLLs
  - Global PLL requires RX clock distribution to individual channels
Clock and Data Recovery

- A clock and data recovery system (CDR) produces the clocks to sample incoming data.
- The clock(s) must have an effective frequency equal to the incoming data rate:
  - 10GHz for 10Gb/s data rate
  - OR, multiple clocks spaced at 100ps
  - Additional clocks may be used for phase detection
- Sampling clocks should have the proper phase relationship with the incoming data for sufficient timing margin to achieve the desired bit-error-rate (BER).
- CDR should exhibit small effective jitter.
Embedded Clocking (CDR)

PLL-based CDR

- Clock frequency and optimum phase position are extracted from incoming data
- Phase detection continuously running
- Jitter tracking limited by CDR bandwidth
  - With technology scaling we can make CDRs with higher bandwidths and the jitter tracking advantages of source synchronous systems is diminished
- Possible CDR implementations
  - Stand-alone PLL
  - “Dual-loop” architecture with a PLL or DLL and phase interpolators (PI)
  - Phase-rotator PLL

Dual-Loop CDR

- Frequency Synthesis PLL
- 5-stage coupled VCO
- 5:1 MUX/Interpolator Pairs
- RX PD early/late
- FSM sel
- 800MHZ Ref Clk
- CP PFD
- $\Phi_{RX}[n:0]$ $\Phi_{PLL}[0]$ $\Phi_{PLL}[4:0]$
CDR Phase Detectors

- A primary difference between CDRs and PLLs is that the incoming data signal is not periodic like the incoming reference clock of a PLL.

- A CDR phase detector must operate properly with missing transition edges in the input data sequence.
CDR Phase Detectors

- CDR phase detectors compare the phase between the input data and the recovered clock sampling this data and provides information to adjust the sampling clocks’ phase.

- Phase detectors can be linear or non-linear.

- Linear phase detectors provide both sign and magnitude information regarding the sampling phase error:
  - Hogge

- Non-linear phase detectors provide only sign information regarding the sampling phase error:
  - Alexander or 2x-Oversampled or Bang-Bang
  - Oversampling (>2)
  - Baud-Rate
Hogge Phase Detector

- Linear phase detector
- With a data transition and assuming a full-rate clock
  - The late signal produces a signal whose pulse width is proportional to the phase difference between the incoming data and the sampling clock
  - A Tb/2 reference signal is produced with a Tb/2 delay
- If the clock is sampling early, the late signal will be shorter than Tb/2 and vice-versa
Hogge Phase Detector

For phase transfer 0 rad is w.r.t optimal Tb/2 (π) spacing between sampling clock and data

\[ \phi_e = \phi_{in} - \phi_{clk} - \pi \]

TD is the transition density – no transitions, no information

A value of 0.5 can be assumed for random data
PLL-Based CDR with a Hogge PD

- XOR outputs can directly drive the charge pump
- Need a relatively high-speed charge pump

[Razavi]
Alexander (2x-Oversampled) Phase Detector

- Most commonly used CDR phase detector
- Non-linear (Binary) “Bang-Bang” PD
  - Only provides sign information of phase error (not magnitude)
- Phase detector uses 2 data samples and one “edge” sample
- Data transition necessary
  \[ D_n \oplus D_{n+1} \]
  - If “edge” sample is same as second bit (or different from first), then the clock is sampling “late”
    \[ E_n \oplus D_n \]
  - If “edge” sample is same as first bit (or different from second), then the clock is sampling “early”
    \[ E_n \oplus D_{n+1} \]
Alexander Phase Detector Characteristic (No Noise)

- Phase detector only outputs phase error sign information in the form of a late OR early pulse whose width doesn’t vary
- Phase detector gain is ideally infinite at zero phase error
  - Finite gain will be present with noise, clock jitter, sampler metastability, ISI
Alexander Phase Detector Characteristic (With Noise)

- Total transfer characteristic is the convolution of the ideal PD transfer characteristic and the noise PDF.
- Noise linearizes the phase detector over a phase region corresponding to the peak-to-peak jitter.

\[ K_{PD} \approx \frac{2}{J_{PP}} (TD) \]

- TD is the transition density – no transitions, no information
  - A value of 0.5 can be assumed for random data.
Mueller-Muller Baud-Rate Phase Detector

- Baud-rate phase detector only requires one sample clock per symbol (bit)

- Mueller-Muller phase detector commonly used

- Attempting to equalize the amplitude of samples taken before and after a pulse

Locked Condition: $h(\tau_k - T_b) = h(\tau_k + T_b)$
Early Clock: $h(\tau_k - T_b) < h(\tau_k + T_b)$
Late Clock: $h(\tau_k - T_b) > h(\tau_k + T_b)$
Mueller-Muller Baud-Rate Phase Detector

Phase Error:
\[ \Delta T_n = D_n \times D_{n-1} \times (\text{ERR}_n - \text{ERR}_{n-1}) \]

Phase detector output truth table

<table>
<thead>
<tr>
<th>( d_j )</th>
<th>( d_{j-1} )</th>
<th>( e_j )</th>
<th>( e_{j-1} )</th>
<th>( \Phi_{err_j} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>LATE</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>LATE</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>EARLY</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>EARLY</td>
</tr>
</tbody>
</table>

All other cases

HOLD

[Spagna ISSCC 2010]
Analog PLL-based CDR

\[ \phi_{out} = \frac{s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}} \]

\[ K_P = I_C \cdot R \quad K_i = \frac{I_C}{C} \quad \omega_n = \sqrt{K_i \cdot K_{PD} \cdot K_{VCO}} \quad \zeta = \frac{K_P}{K_i} \cdot \frac{\omega_n}{2} \]
Analog PLL-based CDR

- CDR “bandwidth” will vary with input phase variation amplitude with a non-linear phase detector
- Final performance verification should be done with a time-domain non-linear model
Single-Loop CDR Issues

- Phase detectors have limited frequency acquisition range
  - Results in long lock times or not locking at all
  - Can potentially lock to harmonics of correct clock frequency
- VCO frequency range variation with process, voltage, and temperature can exceed PLL lock range if only a phase detector is employed
Phase Interpolator (PI) Based CDR

- Frequency synthesis loop can be a global PLL
- Can be difficult to distribute multiple phases long distance
  - Need to preserve phase spacing
  - Clock distribution power increases with phase number
  - If CDR needs more than 4 phases consider local phase generation
DLL Local Phase Generation

- Only differential clock is distributed from global PLL
- Delay-Locked Loop (DLL) locally generates the multiple clock phases for the phase interpolators
  - DLL can be per-channel or shared by a small number (4)
- Same architecture can be used in a forwarded-clock system
  - Replace frequency synthesis PLL with forwarded-clock signals

![Diagram of DLL Local Phase Generation](image)
Phase Rotator PLL

- Phase interpolators can be expensive in terms of power and area.
- Phase rotator PLL places one interpolator in PLL feedback to adjust all VCO output phases simultaneously.
- Now frequency synthesis and phase recovery loops are coupled:
  - Need PLL bandwidth greater than phase loop.
    - Useful in filtering VCO noise.
Phase Interpolators

- Phase interpolators realize digital-to-phase conversion (DPC)
- Produce an output clock that is a weighted sum of two input clock phases
- Common circuit structures
  - Tail current summation interpolation
  - Voltage-mode interpolation
- Interpolator code mapping techniques
  - Sinusoidal
  - Linear
**Sinusoidal Phase Interpolation**

- **Arbitrary phase shift can be generated with linear summation of I/Q clock signal**

\[ X_I = A \sin(\omega t) \]

\[ X_Q = A \sin(\omega t - \pi / 2) = -A \cos(\omega t) \]

\[ Y = A \sin(\omega t - \phi) \]

\[ = A \cos(\phi) \sin(\omega t) - A \sin(\phi) \cos(\omega t) \quad \left(0 \leq \phi \leq \frac{\pi}{2}\right) \]

\[ = \cos(\phi)X_I + \sin(\phi)X_Q = a_1 X_I + a_2 X_Q \]

\[ Y = A \sin(\omega t - \phi) = a_1 X_I + a_2 X_Q \]

where \( a_1 = \cos(\phi) \) and \( a_2 = \sin(\phi) \)

\[ a_1^2 + a_2^2 = 1 \]
Sinusoidal vs Linear Phase Interpolation

- It can be difficult to generate a circuit that implements sinusoidal weighting
  \[ a_1^2 + a_2^2 = 1 \]

- In practice, a linear weighting is often used
  \[ a_1 + a_2 = 1 \]
Phase Interpolator Model

- Interpolation linearity is a function of the phase spacing, $\Delta t$, to output time constant, $RC$, ratio
- Important that interpolator output time constant is not too small (fast) for phase mixing quality
Phase Interpolator Model

w/ ideal step inputs:

\[ V_o(t) = V_{cc} + R \cdot I \cdot \left[ (1 - \alpha) \cdot u(t) \cdot \left( e^{\frac{t}{RC}} - 1 \right) + \alpha \cdot u(t - \Delta t) \cdot \left( e^{\frac{t-\Delta t}{RC}} - 1 \right) \right] \]

w/ finite input transition time:

\[ V_o(t) = V_{cc} + (1 - \alpha) \cdot \frac{I_{\text{max}} \cdot t}{\Delta t} \cdot R \cdot \alpha \cdot [u(t) - u(t - \Delta t)] \cdot \left( e^{\frac{t}{RC}} - 1 \right) + \]

\[ \alpha \cdot \frac{I_{\text{max}} \cdot t}{\tau_r} \cdot R \cdot [u(t - \Delta t) - u(t - 3\Delta t)] \cdot \left( e^{\frac{t-\Delta t}{RC}} - 1 \right). \]

For more details see D. Weinlader’s Stanford PhD thesis
Tail-Current Summation PI

[Bulzacchelli JSSC 2006]

- Control of I/Q polarity allows for full 360° phase rotation with phase step determined by resolution of weighting DAC
- For linearity over a wide frequency range, important to control either input or output time constant (slew rate)
Voltage-Mode Summation PI

[Joshi VLSI Symp 2009]

- For linearity over a wide frequency range, important to control either input or output time constant (slew rate)
Delay-Locked Loop (DLL)

- DLLs lock delay of a voltage-controlled delay line (VCDL)
- Typically lock the delay to 1 or \( \frac{1}{2} \) input clock cycles
  - If locking to \( \frac{1}{2} \) clock cycle the DLL is sensitive to clock duty cycle
- DLL does not self-generate the output clock, only delays the input clock

[Sidiropoulos JSSC 1997]
Voltage-Controlled Delay Line

[Sidiropoulos]
**DLL Delay Transfer Function**

- First-order loop as delay line doesn’t introduce a (low-frequency) pole
- The delay between reference and feedback signal is low-pass filtered
- Unconditionally stable as long as continuous-time approximation holds, i.e. $\omega_n < \omega_{ref}/10$

\[
D_O(s) = (D_I(s) - D_O(s)) \cdot F_{REF} \cdot \frac{I_{CH}}{sC_1} \cdot K_{DL}
\]

\[
\frac{D_O(s)}{D_I(s)} = \frac{1}{1 + s/\omega_N}
\]

\[
\omega_N = I_{CH} \cdot K_{DL} \cdot F_{REF} \cdot \frac{1}{C_1}
\]
CDR Jitter Properties

- Jitter Transfer
- Jitter Generation
- Jitter Tolerance
CDR Jitter Model

\[
\frac{\phi_{out}}{\phi_{in}} = \frac{s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}
\]

\[
K_P = I_C \cdot R \quad K_i = \frac{I_C}{C} \quad \omega_n = \sqrt{K_i \cdot K_{PD} \cdot K_{VCO}} \quad \zeta = \frac{K_P}{K_i} \cdot \frac{\omega_n}{2}
\]
Jitter Transfer

- Jitter transfer is how much input jitter “transfers” to the output
  - If the PLL has any peaking in the phase transfer function, this jitter can actually be amplified

\[
\frac{\phi_{\text{out}}}{\phi_{\text{in}}} = \frac{s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}
\]
Jitter Transfer Measurement

\[ JTF(f) = 20 \log \left( \frac{\text{Output Jitter}(f)}{\text{Input Jitter}(f)} \right) \]

[TrV89] [RaO91]

[Walker]
Jitter Transfer Specification

![Diagram showing Jitter Transfer Specification with a slope of -20 dB/decade and acceptable range.]

This specification is intended to control jitter peaking in long repeater chains.

<table>
<thead>
<tr>
<th>Data Rate</th>
<th>( f_c ) [kHz]</th>
<th>( P ) [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>155 Mb</td>
<td>130</td>
<td>0.1</td>
</tr>
<tr>
<td>622 Mb</td>
<td>500</td>
<td>0.1</td>
</tr>
<tr>
<td>2.488 Gb</td>
<td>2000</td>
<td>0.1</td>
</tr>
</tbody>
</table>

[Walker]
Jitter Generation

- Jitter generation is how much jitter the CDR “generates”
  - Assumed to be dominated by VCO
- Assumes jitter-free serial data input

VCO Phase Noise:

\[ H_{\nu_{VCO}}(s) = \frac{\phi_{out}}{\phi_{\nu_{VCO}}} = \frac{s^2}{s^2 + \left( \frac{K_{Loop}}{N} \right) RCs + \frac{K_{Loop}}{N}} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \]

For CDR, N should be 1
Jitter Generation

High-Pass Transfer Function

\[ 20 \log_{10} \left( \frac{\theta_{\text{out}}(s)}{\theta_{\text{con}}(s)} \right) \]

Jitter accumulates up to time \( \propto \frac{1}{\text{PLL bandwidth}} \)

- SONET specification:
  - rms output jitter \( \leq 0.01 \text{ UI} \)

[McNeill]
Jitter Tolerance

- How much sinusoidal jitter can the CDR “tolerate” and still achieve a given BER?

Maximum tolerable $\phi_e$

$$
\phi_e(s) = \left(1 - \frac{\phi_{out}(s)}{\phi_{in}(s)}\right) \phi_{in}(s) \leq \frac{\text{Timing Margin}}{2}
$$

$$
JTOL(s) = 2\phi_{in}(s) = \frac{TM}{\left(1 - \frac{\phi_{out}(s)}{\phi_{in}(s)}\right)}
$$

[Sheikholeslami]

[Lee]
Jitter Tolerance Measurement

- Random and sinusoidal jitter are added by modulating the BERT clock
- Deterministic jitter is added by passing the data through the channel
- For a given frequency, sinusoidal jitter amplitude is increased until the minimum acceptable BER ($10^{-12}$) is recorded
Jitter Tolerance Measurement

\[ JTOL(s) = 2\phi_{n.in}(s) = \frac{TM}{1 - \frac{\phi_{out}(s)}{\phi_{in}(s)}} \]

Flat region is beyond CDR bandwidth

(within CDR bandwidth)

[Lee]
Next Time

- Forwarded-Clock Deskew Circuits
- Clock Distribution Techniques