EE290C - Spring 2004
Advanced Topics in Circuit Design
High-Speed Electrical Interfaces

Lecture #15

CDR & Coding

Jared Zerbe
3/9/04

Agenda

- Introduction
- Example CDRs
- CDR Issues
- Second-order loops
- 4-PAM CDRs
- Coding and Transition Density
Clocking: Terminology

- **Synchronous**
  Every participant gets same frequency and phase.

- **Mesochronous**
  Every participant gets same frequency, but unknown phase. Requires a way to recover the phase from the data. Coding (e.g. 8b/10b) is often used to make sure there are sufficient data edges.

- **Plesiochronous**
  Every participant gets nearly the same frequency, slowly drifting phase. Requires a way to detect when the Rx clock has drifted 1/2 cycle from Tx clock.

- **Asynchronous**
  Dispense with clocks altogether, use (e.g.) request/acknowledge 4-phase handshake to ensure correct sequencing of events.

What's a CDR?
Clock and Data Recovery

- **Recovering clock from the data**
  - Can recover clock completely, or just phase
  - Just phase: need a reference clock

- **Why?**
  - Allows separate xtals on different boards
  - Don't have to match trace lengths, delays
  - Easier system design / clock distribution

- **Why Not**
  - Expensive: takes area, power
  - Requires coding or transition density or at a minimum a training sequence
    - 8b10b coding uses 10b to xfer 8b of info; 20% BW loss
Clock Data Recovery
Clock Data Recovery

Agenda

- Introduction
- Example CDRs
- CDR Issues
- Second-order loops
- 4-PAM CDRs
- Coding and Transition Density
Example CDR: PLL Technique

- Simple bang-bang PLL
  - Observe data with phase detector
  - Filter Early/Late & drive VCO
- Advantages
  - Good frequency range
  - Low Jitter
- Challenges
  - Phase offset
  - Lock time - startup sequence
  - Loss of lock - coding dependant
  - How to integrate?
    - Multiple PLLs
    - Harmonic locking problems

Dual PLL Problem: Harmonic Locking

- Potentially serious in highly integrated plesiochronous systems where residual phase error is close to noise injection in magnitude
2-PAM Eye With Density

2x Oversampling

- **Generate early/late from** $d_n, d_{n-1}, e_n$
  - Simple 1st order loop, cancels receiver setup time
- **Jitter on** $d$ Clk ≠ PLL output
  - Base is linear PLL jitter
  - Can add non-linear phase selector noise from CDR
Dual-Loop CDR

- Combination of
  - Core PLL provides multiple phases at frequency
  - Periphery DLL mixes and make desired phase

- Advantages
  - Avoids harmonic locking
  - Easy to integrate many
  - Rapid CDR lock time
  - CDR very stable
    - Digital = flexible filtering, control
    - Can even ‘hold’ phase state

- Challenges
  - Limited Freq offset from PLL
  - Jitter not as low as PLL

Baud-Rate CDR

- Use information from data-level sampler (already there for adaptation) : eliminate edge sampler, eclk
  - Curvature to waveform here
    - Use a comparitor to differentiate between dlev & signal
    - Decide later if 0 or 1 from comparator means early or late
    - Even more transition-like level behavior in 4-PAM mid-data levels
**Agenda**

- Introduction
- Example CDRs
- CDR Issues
- Second-order loops
- 4-PAM CDRs
- Coding and Transition Density

---

**CDR Issues : Transitions & Tracking**

- CDRs require transition density
  - PLL based CDRs require to keep lock even in mesochronous
  - DLL based require for plesiochronous tracking
- Often coding is used to guarantee
- Can alternately use a scrambler + XOR
CDR Issues: Jitter

- CDR Jitter starts out worse than PLL jitter
- Also can have ‘dither jitter’: phase wander when locked

CDR Dither Jitter

- Caused by need to track plesiochronous differences
- Dither jitter set by
  - Latency of the loop: usually 10-20 cycles
  - Step size: usually 10m-UI
  - # of averages: usually 16+

- The last two along with transition density set the tracking rate
  - -> Conflicting requirements between tracking and jitter
Agenda

- Introduction
- Example CDRs
- CDR Issues
- **Second-order loops**
- 4-PAM CDRs
- Coding and Transition Density

Second-Order Loops

Idea: solve the plesiochronous vs. jitter problem

- Build a second-order loop instead
- One loop for tracking the constant frequency difference with very low tracking rate & bandwidth (thus low jitter)
- A very small step size & dither for tracking normal phase shifts

- Currently an area of interesting research
Agenda

- Introduction
- Example CDRs
- CDR Issues
- Second-order loops
- 4-PAM CDRs
- Coding and Transition Density

4-PAM: Multiple Transitions

» So many transition types make 2-PAM CDR unusable
4-PAM Eye With Density

- Offset transitions clearly visible
- But good transitions exist...

4-PAM Edges & CDR Approach #1

- Offset edge sampler to data level to get mid-levels
  - Best available edge-rate in FSE system
  - Requires edge samplers placed accurately on data level
4-PAM Edges & CDR Approach #2

- Use all minor transitions and one major transition
- More transitions to choose from, no voltage offset required
- Poorer edge-rate from minor transitions

Measured 4-PAM CDR Performance

- 2-PAM CDR on 4-PAM data
  - 60ps p-p @ 8Gb/s
- 4-PAM CDR uses only minor transitions
- Lower dither jitter
  - 35ps p-p @ 8Gb/s
**Agenda**

- Introduction
- Example CDRs
- CDR Issues
- Second-order loops
- 4-PAM CDRs
- Coding and Transition Density

---

**CDR In a Plesiochronous System**

**CHIP 1**
- Encode
- Serializer
- Tx
- 8b/10b
- f1
- PLL
- 8 bits @ f1

**CHIP 2**
- Rx
- Deserializer
- FIFO
- Elastic Buffer
- CDR
- 10b/8b
- f2
- 10 bits @ f1
- 10 bits @ f2
- 8 bits @ f2

Goal is to transfer 8 bits @ f1 on chip 1 to 8 bits @ f2 on chip 2
- First encode and transfer data based on local clock f1
- Then recover data and clock (f1) on chip 2
- Elastic buffer (FIFO) used to transfer data from f1 to f2
- Finally, decode to get 8 bits @ f2
Plesiochronous System Impact

- There must be packets with appropriate slack time
  - How else to recover timing difference?
  - Idle characters must be recognized for slack
- There must be FIFOs deep enough
  - Set by maximum frequency difference & maximum data length
- CDR must have tracking rate
  - Must be able to track maximum difference including dither
- Frequency difference, protocol, maximum data packet size different system components

CDR : Coding & Transition Density

- CDR requires transition density to keep lock
- AC - coupling requires DC-balance
- Plesiochronous operation requires packets & null characters
  - Coding as solution
- Typical code : 8b10b from IBM
  - 8 bits into the link => 10 bits on the wires
  - Raw data rate must be 25% faster than effective data rate
    - 6.25Gb raw for 5Gb effective
  - 8b10b code guarantees
    - DC balance
    - Transition density : 2 transitions every 10 bits
    - Reserved codes, control characters
**8b10b Code Overview**

- DC Balanced within every code word

**64/66 Code Overview**

Data Codewords have "01" sync preamble

<table>
<thead>
<tr>
<th>0 1</th>
<th>64 bit data field (scrambled)</th>
</tr>
</thead>
</table>

Mixed Data/Control frames are identified with a "10" sync preamble. Both the coded 56-bit payload and TYPE field are scrambled

<table>
<thead>
<tr>
<th>1 0</th>
<th>8-bit TYPE</th>
<th>combined 56 bit data/control field (scrambled)</th>
</tr>
</thead>
</table>

00,11 preambles are considered code errors and cause the packet to be invalidated by forcing an error (E) symbol on the HARI output

- Much lower overhead
- Poorer DC balance & transition properties