Clock Recovery Lectures

Applications

Clock extraction from NRZ data

Phase detection and data retiming

Bang-Bang
Hagge
Alexander (early-late)

Frequency Detection

Combine FD and PD for CDR

References:


Diversity of CDR applications

- Clock and Data Recovery applications span the range from high-volume, low-cost datacom applications to high-performance, long-haul telecom applications

- Many different trade-offs tailor each circuit to the target

Basic Idea

Serial data transmission sends binary bits of information as a series of optical or electrical pulses:

The transmission channel (coax, radio, fiber) generally distorts the signal in various ways:

From this signal we must recover both clock and data.
NRZ and RZ signalling

NRZ = “non return to zero” data

RZ = “return to zero” data

NRZ signalling is almost universally used.

Walker, op.cit.
Spectrum of NRZ data

variations due to DC balance strategy

missing clock frequency

\[
\frac{\sin(2\pi fT)}{2\pi fT}
\]

\( F_c - \delta \)

\( F_c + \delta \)

\( f = 0 \)

\( 1/T \)

\( 2/T \)

Walker, op. cit.
Clock and Data Recovery

Extract a clock signal from binary data

1. Data Format: NRZ
   - If one bit period is \( T_b \)
   - Bit rate is \( \frac{1}{T_b} \)
   - i.e. 1 ns \( \rightarrow \) 1 Gb/s

2. Fastest Data Sequence for NRZ: 101010
   - Square wave @ \( f = \frac{1}{2T_b} \)
   - \( T_b \) bit

No freq. components at even harmonic!
This makes clock extraction more complicated.

3. In addition, it's possible to have long strings of '1' or '0'.
   - CRC must remember the bit rate while no input is received.
Clock and Data Recovery

The incoming signal will always have some noise. NL fiber effects can also cause ISI. To get highest SNR or best BER, we need to sample the data at the right time.

Data looks like pseudorandom NRZ format:

\[
\begin{align*}
\left< T_b \quad \left< \text{jitter} \quad \left< \text{Data} \quad \left< \text{Clock}
\end{align*}
\]

Need to have a clock source:
1. Right frequency
2. Right phase
How to address this problem.

**Edge Detection.**

If both positive and negative transitions are detected, we can produce a freq. component at the clock rate.

![Diagram of edge detection circuit](image)

This addresses the first problem -

The PLL or DLL is used to address the second problem - long stretches of 1 or 0 data.

![Diagram of PLL circuit](image)

The LFF time constant must be long enough to hold the VCO frequency constant during such a gap in input. This slows acquisition 😊
Figure 9.7 (a) Edge detector circuit. (b) Simplified circuit for rising data edge. (c) Simplified circuit for falling data edge.

Razavi, op.cit.
Figure 9.8  (a) Digital edge detector, (b) circuit implementation of (a).

Phase Detectors. Behave differently with random data since data contains no component of frequency at the clock rate.

XOR:

Think of it as a multiplier

\[ V_{\text{out}}(+) = D_{\text{in}}(+) \cos \omega_c t \]

\[ \omega_c = \frac{2\pi}{T_b} \]

when we average (integrate) two sinusoidal signals of different frequencies, the average value = 0.

or, if we have equal probability of ONE and ZERO in data stream, multiplication by a sine wave at \( \omega_c \) averages to zero.
Figure 9.10 Failure of XOR gate as a phase detector with random data.

Razavi op. cit.
Phase Detectors.

DEF. if data sampled by clock, and equal probability of “1” and “0”, average is zero.

if clock is sampled by data, we have valid phase detection.

<table>
<thead>
<tr>
<th>Data lags clock</th>
<th>Output = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data leads clock</td>
<td>Output = 0</td>
</tr>
</tbody>
</table>

bang-bang PD.

\[
\begin{align*}
V_{out} & \quad \Delta \phi \\
\uparrow & \quad \text{very high gain}
\end{align*}
\]
Sec. 9.1 General Considerations

Figure 9.11 Use of a D flipflop as a phase detector: (a) data sampled by clock, (b) clock sampled by data.

Ratatvī. op. cit.
In addition, since DFF is edge-triggered, we get sampling only on single data edge.
This is equivalent to differentiating the data and using only one edge, non-zero output.

Invert Clock

But as the PD drives the phase error toward zero,
the DFF runs with metastability. This limits its effective gain.

Capture, fVCO and fDATA are different
beat note @ fVCO - fDATA
drives PLL toward lock, if within LPF gain.

When locked, zero crossings of clock are sampled.
dithering problem with long seq. of 1s or 0s
Simple latch model.

\[ \tau_0 = RC \]

Represent as single dominant pole for determining transient response.

\[ \frac{N_x - N_2}{R} - C \frac{dN_2}{dt} = 0 \]

\[ -N_2 - \tau_0 \frac{dN_2}{dt} = -N_x \]

\[ N_2 = \frac{-N_y}{A_0} \]

\[ \frac{N_y}{A_0} + \frac{\tau_0}{A_0} \frac{dN_y}{dt} = -N_x \]

\[ N_y + \tau_0 \frac{N_y}{dt} = -A_0 V_x \]
Likewise,

\[ N_x + T_0 \frac{dN_x}{dt} = -A_0 N_y \]

Subtract

\[ \frac{T_0}{d} \frac{d (N_x - N_y)}{dt} = -(1 - A_0) (N_x - N_y) \]

Initial condition \( N_{xy} = (N_x - N_y) \bigg|_{t=0} \)

\[ N_x - N_y = N_{xy} \exp \left[ (A_0 - 1) \frac{t}{T_0} \right] \]

growing with time since \( A_0 > 1 \)

**BUT**

with time constant \( T_0 / (A_0 - 1) \)

and if \( N_{xy} \) is very small, it could take a long time to reach valid logic levels.
We have skew coming from unequal CK→Q and S→Q delay through PD and re-timing FFs.

This can skew the sampling point.

(re-timing)

(ramdon)

FF kickback can cause VRO jitter.
Clock Recovery

Purpose:
- recover clock from random data sequence
- adjust phase relationship so that data can be retimed reliably
- reduce jitter through filtering of PLL or DLL and resampling with FF.

Data Eye: \( \rightarrow \frac{T_b}{2} \rightarrow \)

Data

Clock

Ideal sampling point is at \( \frac{T_b}{2} \), centered between data edges.

"Phase Margin" = 180° in this case because clock can drift ±180° for ideal clock and data where rise and fall times << \( T_b \).
But, timing errors due to phase offset and jitter on the data or clock can reduce this margin.

(Note that this definition of phase margin is totally unrelated to the feedback system definition.)

Amplitude noise can also lead to sampling error. A "0" or a "1" can be misinterpreted.

These effects cause the Bit Error Rate to increase.
We saw that the DFF could serve as a phase detector for random data.

1. In a PLL, the DFF PD is clocked by the data. The retiming FF is clocked with a 180° phase-shifted clock.

   \[ D \rightarrow q \text{ delay} \]

   \[ \text{delays are often different} \]

   This leads to a phase error in the PD corresponding to the delay difference \( \Delta T \).

   Similar delay mismatch in retiming FF reduces the phase margin.

2. If we get long sequence of 0 or 1 at data input, the DFF will produce a constant output. This will cause the phase to drift and could produce bit errors unless the PLL loop time constant is long. Even if errors are not generated overnight, the "bang-bang" PD causes jitter.
So, a better PD is desired for random data.

Need PD to
1. detect data transitions
2. detect phase difference
3. retime data within the PD
4. minimize skew

Must use VCO output to retime data.

PD should be driven by the clock, not the data.

Fig 9.20. Use FF to delay data.
- output XOR pulse width is proportional to phase offset ⇒ linear
- a pulse is produced at each data edge ⇒ edge detection
- node B gives resampled data

BUT

Compare 9.20 (b) and (c).

Pulse width is also pattern dependent.
Thus, a false locking will occur.

Need to modify →
This ambiguity can be resolved if the proportional pulses from \( y \) are also compared against reference pulses with fixed width \( T_0/2 \). These must also appear only on data edges.

**Fig. 9.21** Hogge PD

Retimed data can be obtained from either \( B \) or \( A \), however, \( A \) is shifted by \( T_0/2 \).

But, there is still a problem or two...

1. \( \text{clk to } A \) delay in \( FF1 \) widens the pulse at \( Y \) by \( \Delta T \).
   \( \text{clk} \rightarrow Q \) of \( FF2 \) just shifts the ref. pulse, width is the same.

So, \( Dn \) and \( cle \) maintain a skew by \( \Delta T \). This reduces the phase margin. Especially important for high data rates.
Figure 9.21 (a) Hogge phase detector and (b) its waveforms.

\[ Y - X \]

\[ \int (Y - X) dt \]
2. Itogge also has a problem with the Tek/2 skew between X and Y. The ref pulse occurs after the proportional pulse.

Fig 9.25

when X and Y drive a charge pump, we get a triangle output

\[ \int (y-x) \, dt \neq 0 \]

Y=up  X=down

This produces a net phase offset

modifications can compensate for the offset

Fig 9.26.

when loop is locked, charge pump output remains constant, even with long strings of 1 or 0. Better than bang-bang DFF PD which generates a lot of jitter for random data sequences.
Figure 9.25 (a) Hogge PD waveforms, (b) resulting triwave produced by a charge pump.

Figure 9.26  Modified Hogge PD to remedy the triwave issue.

Figure 9.27  CDR circuit using Hogge PD.

Alexander or Early-Late PD.

Fig 9.28

Multiple points are sampled.

Din

\[ \begin{array}{ccc}
\text{ck} & \Uparrow & \Uparrow \\
S_1 & S_2 & S_3
\end{array} \]

\[ \begin{array}{c}
110 \\
001 = \text{early} \\
011 = \text{late} \\
100 \\
000 = \text{no transition} \\
111 = \text{do nothing}
\end{array} \]

\[ \begin{array}{ccc}
S_1 \oplus S_2 \text{ and } S_2 \oplus S_3
\end{array} \]

\[ \begin{array}{c}
late \\
early \\
? \text{ no data}
\end{array} \]

Fig. 9.29

X and Y outputs drive a charge pump or when converted to current, can be subtracted.

Fig 9.31.
Figure 9.28 (a) Three-point sampling of data by clock, (b) Alexander phase detector, and (c) its waveforms.

Figure 9.29  Alexander PD waveforms for (a) late and (b) early clock.

Figure 9.31  CDR circuit using Alexander PD.

Net result is zero output when CDR is locked. Output holds its value when there is an absence of data transitions.

Data is recovered at the outputs of FF1 and FF2.

Frequency Detection.

Capture range of PLL will be too small to accommodate process and temperature variations. If $f_{vco}$ is too far from $f_{data}$, loop will not lock.

Thus, some assistance is needed to force $f_{vco}$ to the data frequency.

FD output is needed that gives the polarity of the frequency error.
Digital FD.

Uses double edge-triggered DFF clocks on both rising and falling data edges.

Quadrature clock generation is necessary

If data freq. ≠ fVCO, sampling points on CKI and CKQ will drift. Transition polarity of X4 and X0 can be used to produce bang-bang FD.
Sec. 9.3 Frequency Detectors for Random Data

Figure 9.43 FD waveforms for fast and slow clocks.

the result is high and for \( f_{VCO} < R_b \), it is low. Illustrated in Fig. 9.44 [12], the overall

Figure 9.44 Complete frequency detector.

In the CoC the two control loops must be combined in such a way that they do not fight with each other.

Fig. 9.47.
- VCO is configured with coarse and fine controls.
- Since \( k_{VCO} \) may be very high in systems with low \( V_{DD} \), ripple can be reduced by using a fine input for phase control.
- FO input may be disabled after PD locks to reduce possible jitter.

Fig. 9.48. Dual VCOs.

\( VCO_2 \), identical to \( VCO_1 \), is used for frequency detection. Generates coarse control voltage for PLL.

But, external reference is required. There will be some freq. error. Loop 1 must have adequate range to correct \( VCO_1 \).
Figure 9.47  CDR architecture with coarse and fine VCO control.

Figure 9.48  CDR architecture using two VCOs.