Clock Recovery Lectures

Applications

Clock extraction from NRZ data

Phase detection and data retiming

- Bang-Bang
- Hogge
- Alexander (early-late)

Frequency Detection

Combine FD and PD for CDR

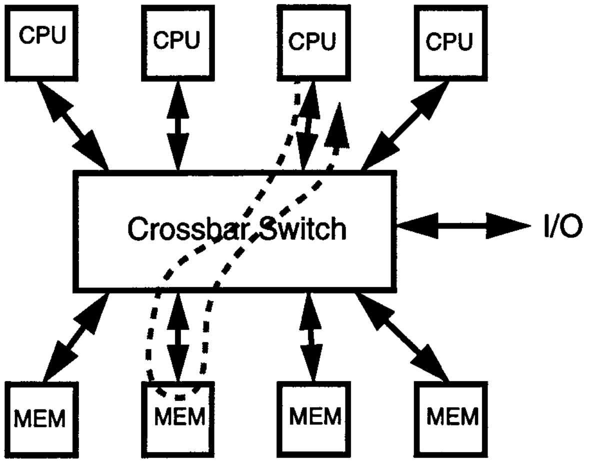
References.

B. Razavi, Design of ICs for optical communications, McGraw-Hill, 2003. Chap. 9.

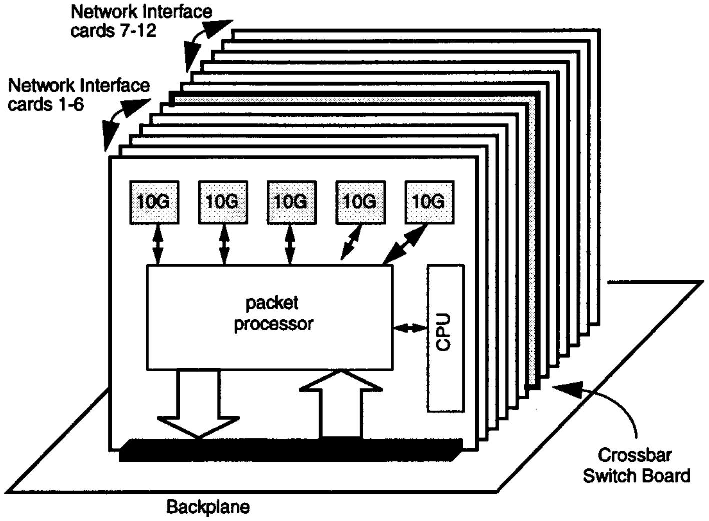
R. Walker, Wideband Communication Workshop, IEEE ISSCC 2002.

Phase-Locking in High-Performance Systems. IEEE Press, 2003.

Diversity of CDR applications



Linecard to Router



CPU-CPU/Memory

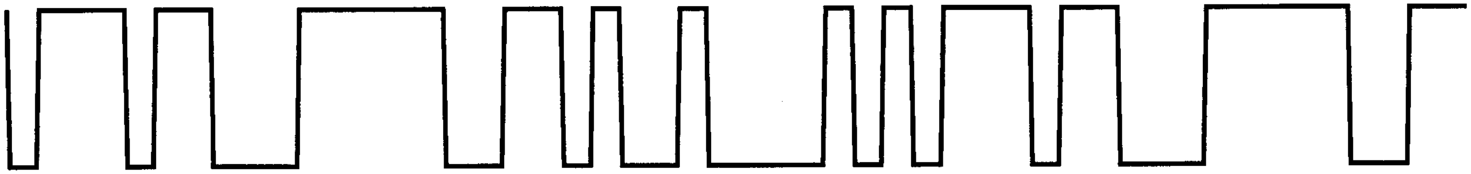
- Clock and Data Recovery applications span the range from high-volume, low-cost datacom applications to high-performance, long-haul telecom applications
- Many different trade-offs tailor each circuit to the target

R. Walker, Wideband Communication Workshop, ISSCC 2002.

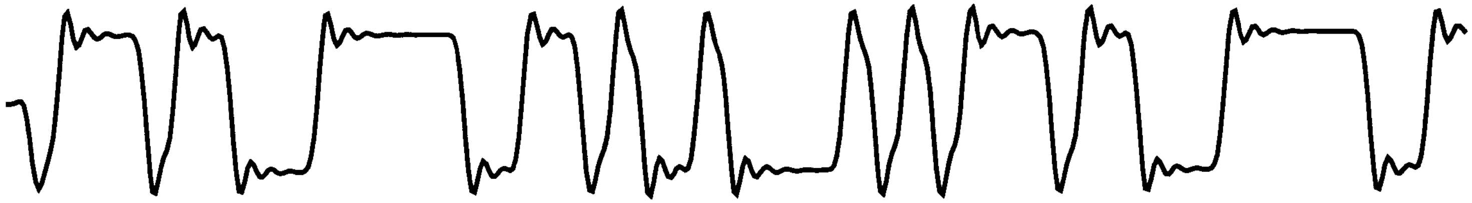
Basic Idea

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

0111011000111110011010010000101011101100011111..



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

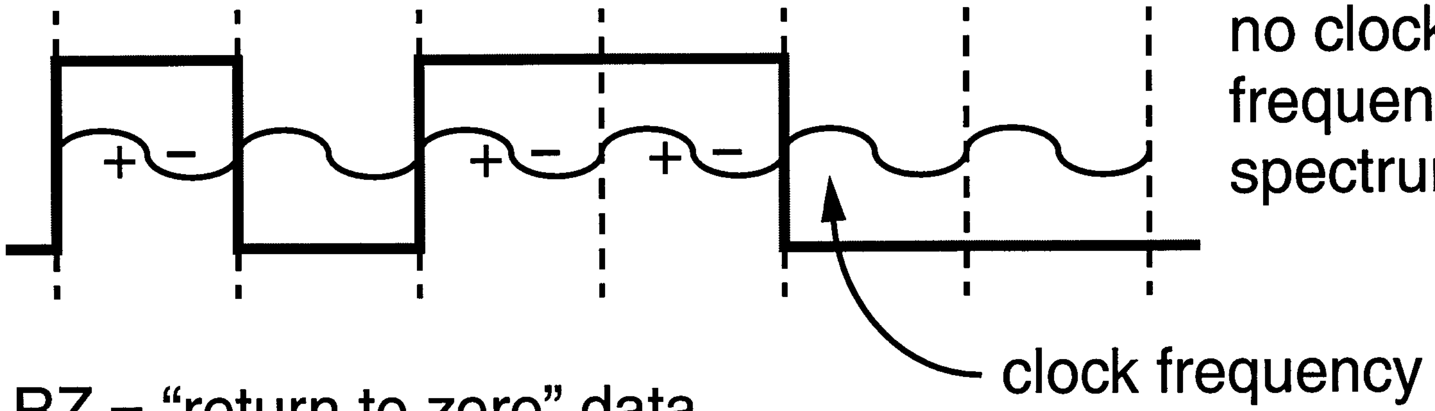


From this signal we must recover both clock and data

R. Walker, op. cit.

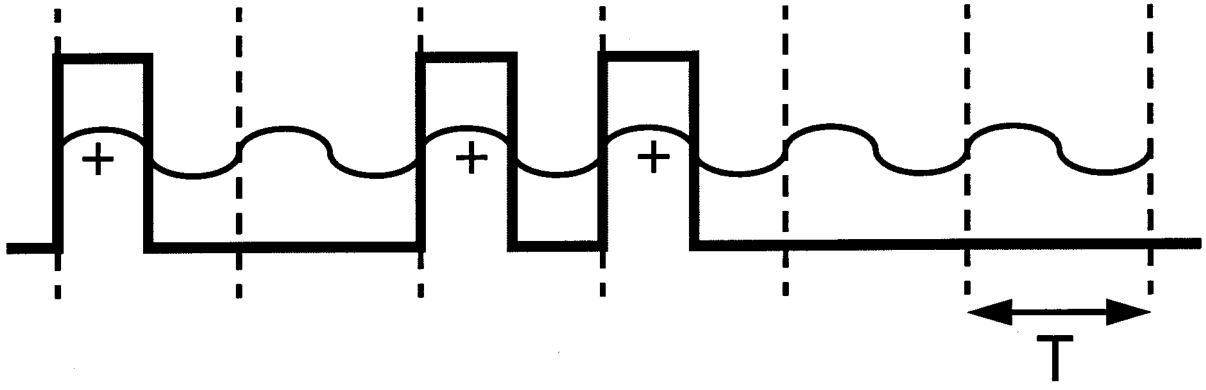
NRZ and RZ signalling

NRZ = "non return to zero" data



no clock
frequency in
spectrum

RZ = "return to zero" data

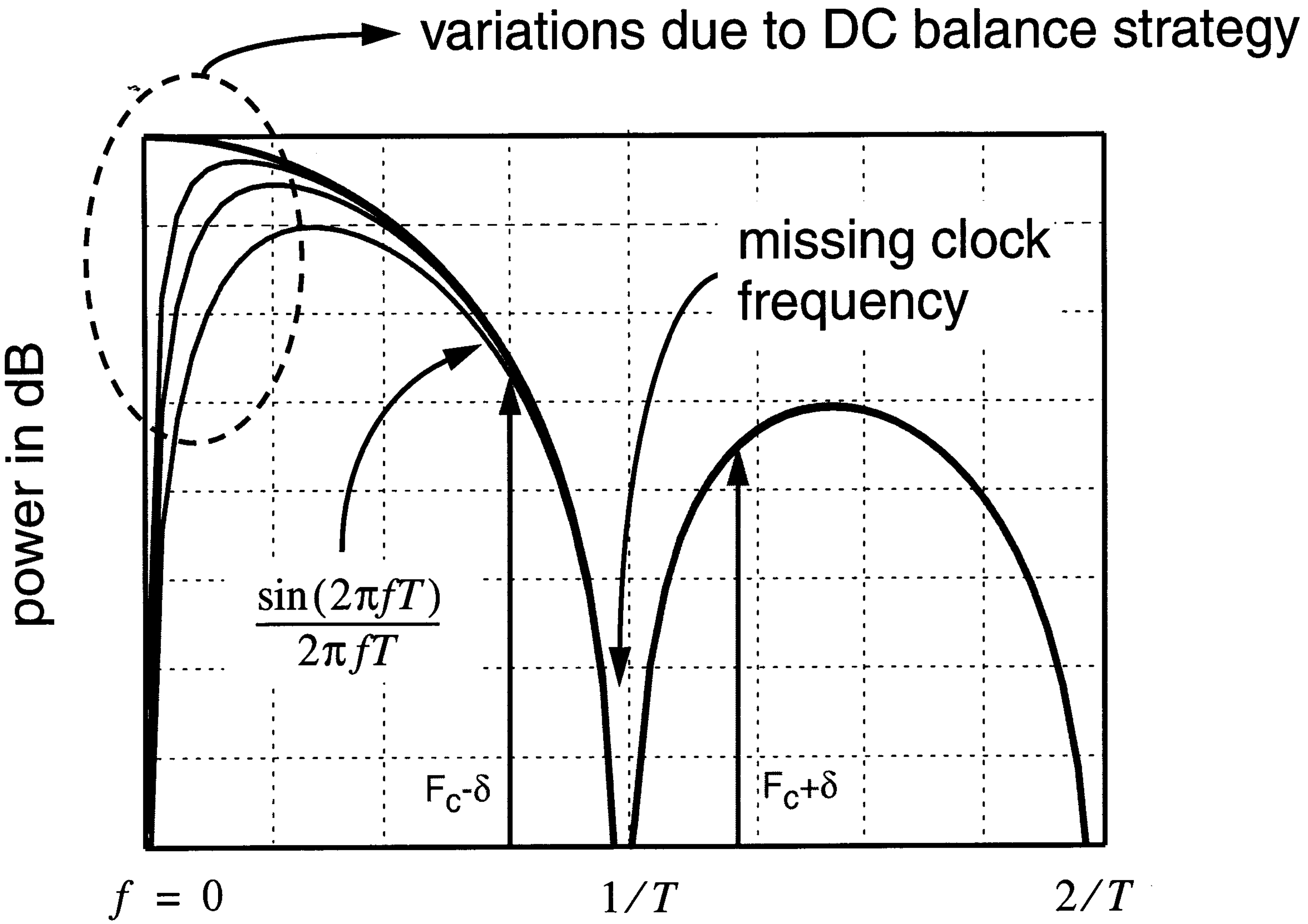


clock frequency
appears in
spectrum

NRZ signalling is almost universally used.

Walker, op.cit.

Spectrum of NRZ data



variations due to DC balance strategy

missing clock frequency

Walker, op. cit.

Clock and Data Recovery

Extract a clock signal from a binary data

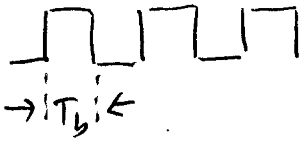
1. Data Format: NRZ

   if one bit period is T_b

   bit rate is $1/T_b$

   ie. 1 ns → 1 Gb/s

2. Fastest Data Sequence for NRZ: 101010



square wave @ $f = 1/2T_b$

No freq. components at even harmonic!

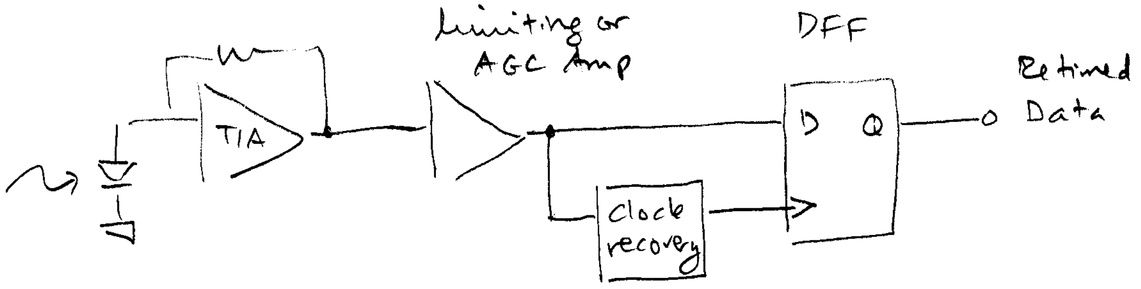
This makes clock extraction more complicated.

3. In addition, its possible to have long strings of '1' or '0'.

   CRC must remember the bit rate while no input is received.

Clock and Data Recovery

O/E Receiver

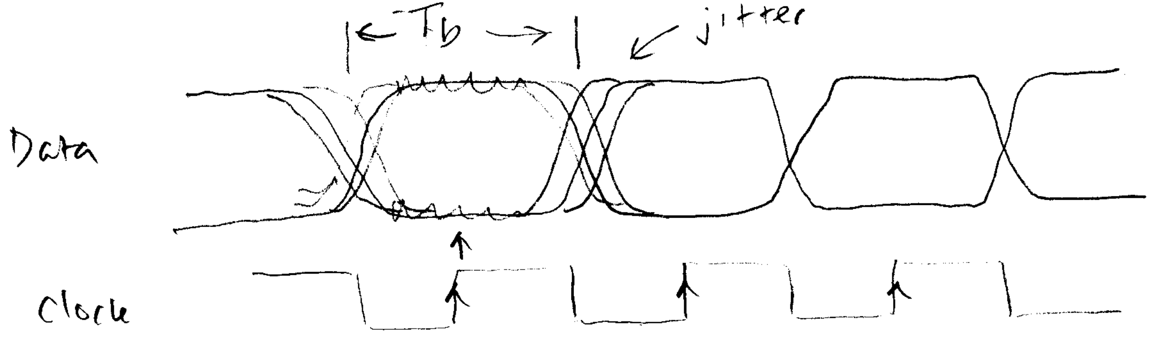


The incoming signal will always have some noise

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



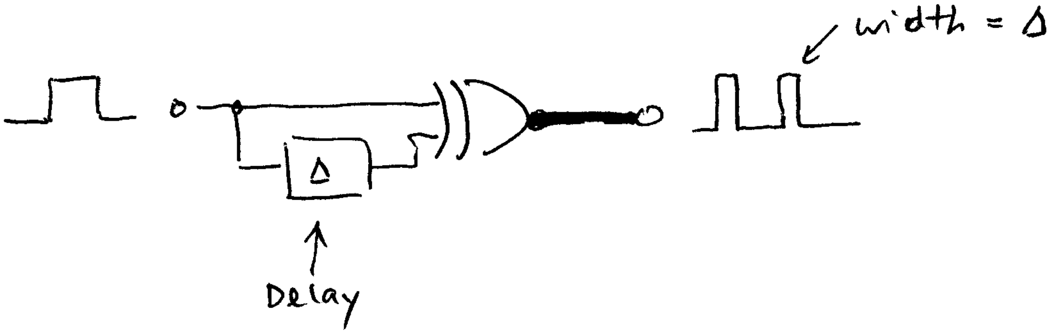
Need to have a clock source

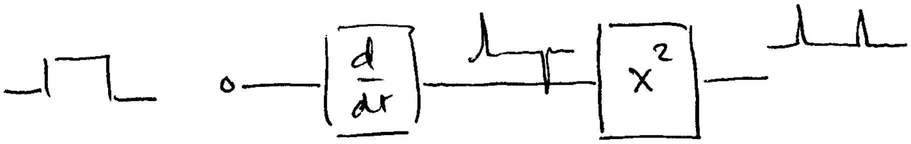
1. right frequency
2. right phase

How to address this problem.

<u>Edge Detection.</u>

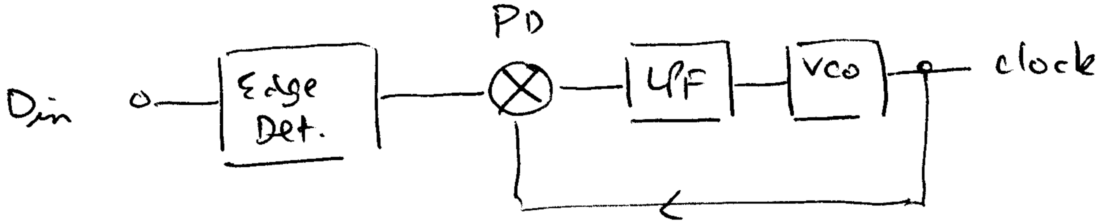
If both positive and negative transitions are detected, we can produce a freq. component at the clock rate.





This addresses the first problem -

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



The LPF time constant must be long enough to hold the VCO frequency constant during such a gap in input. This slows acquisition ☹

The level shift stages

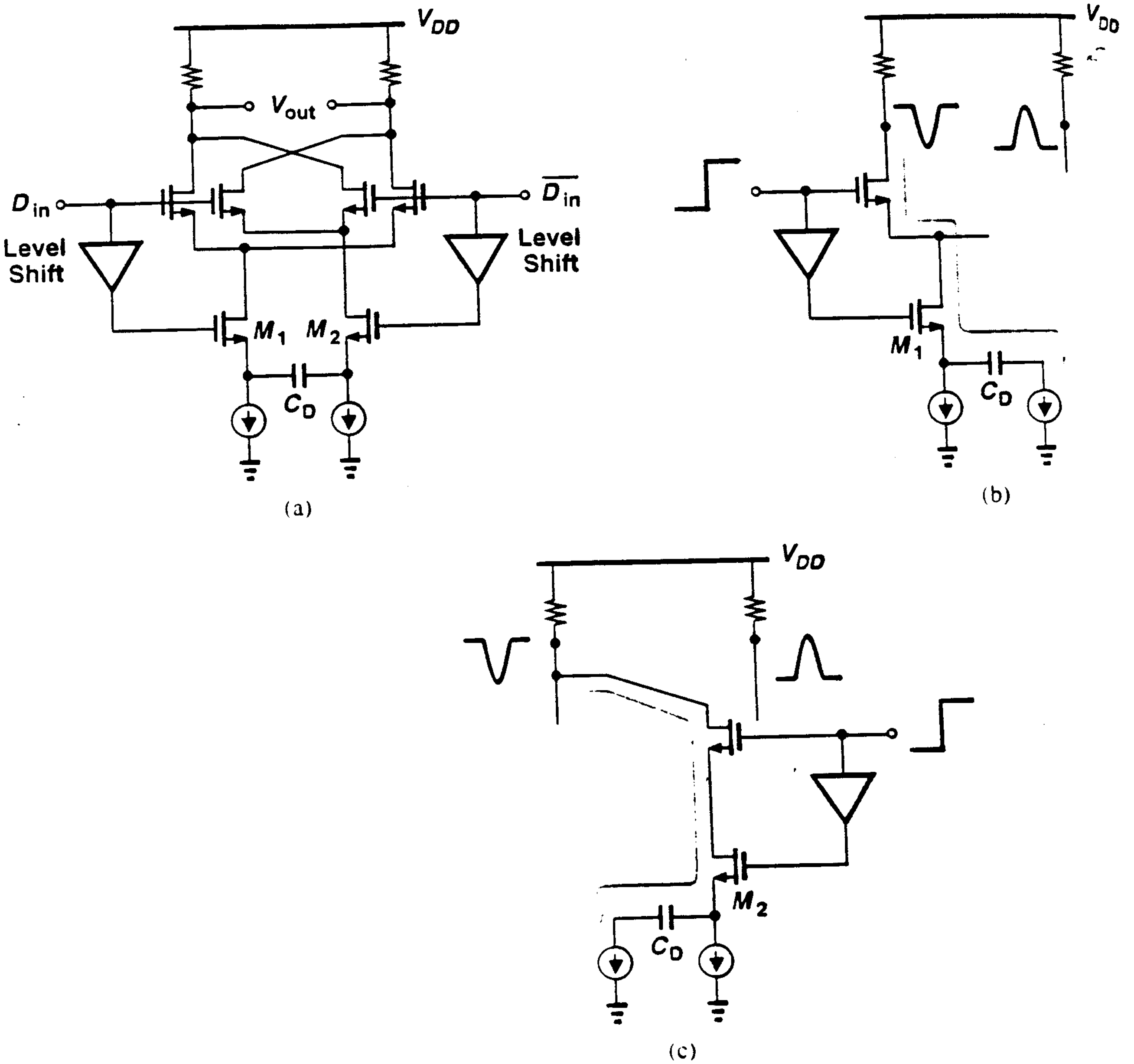


Figure 9.7 (a) Edge detector circuit, (b) simplified circuit for rising data edge, (c) simplified circu for falling data edge.

Razavi. op. cit.

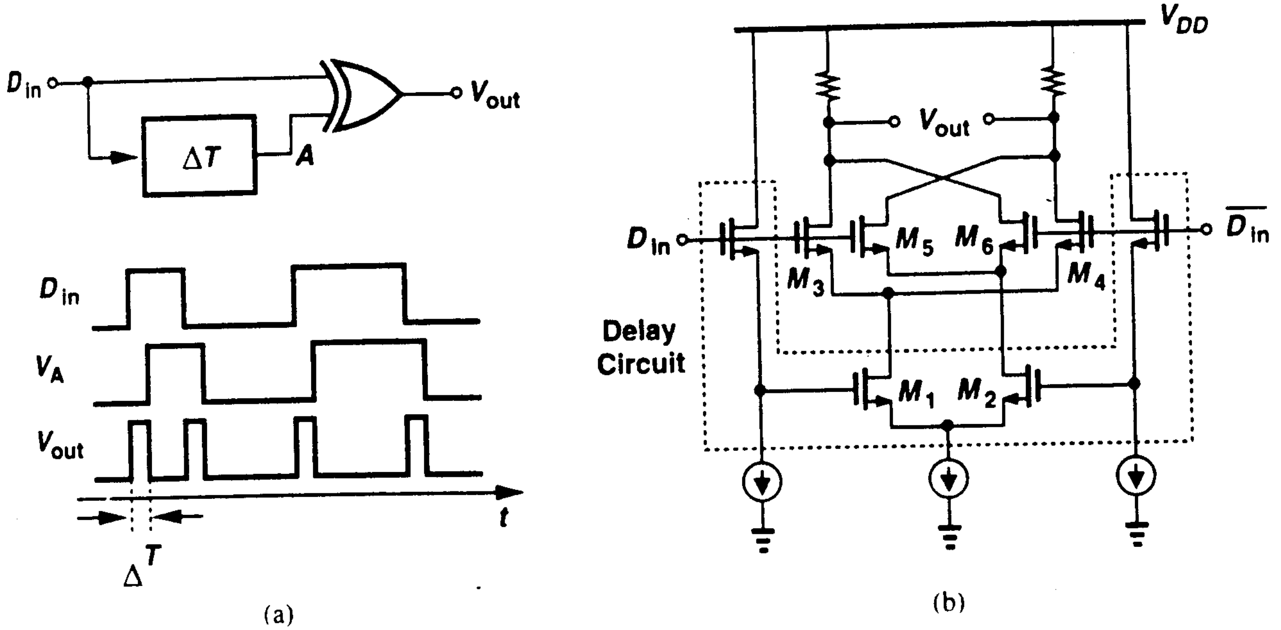


Figure 9.8 (a) Digital edge detector, (b) circuit implementation of (a).

Razavi, Design of Integrated Circuits for Optical Communications,
McGraw-Hill, 2003.

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

$$V_{out}(t) = D_{in}(t) \cos \omega_{clk} t \qquad \omega_{clk} = \frac{2\pi}{T_b}$$

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

or, if we have equal probability of ONE and ZERO in data stream, multiplication by a sine wave at ω_{CLK} averages to zero.

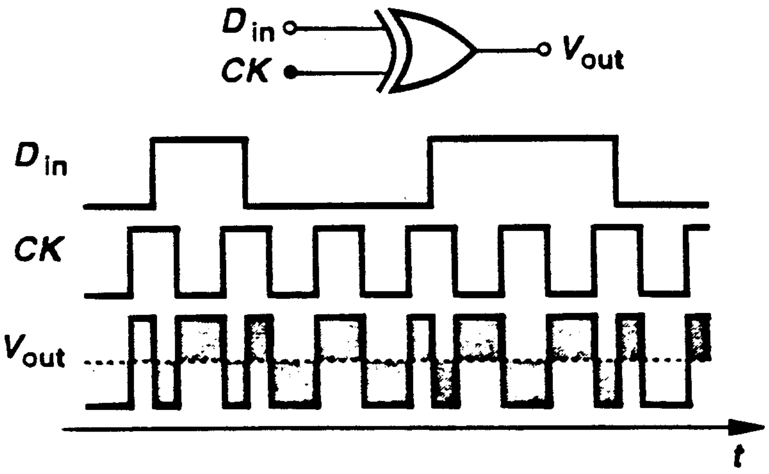


Figure 9.10 Failure of XOR gate as a phase detector with random data.

Razavi op. cit.

Phase Detectors.

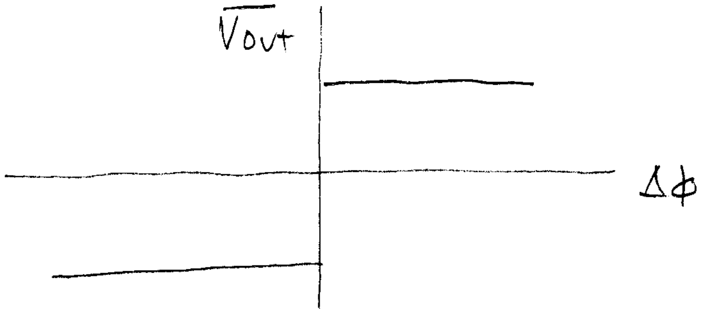
DFF. if data sampled by clock, and equal probability of "1" and "0", average is zero.

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

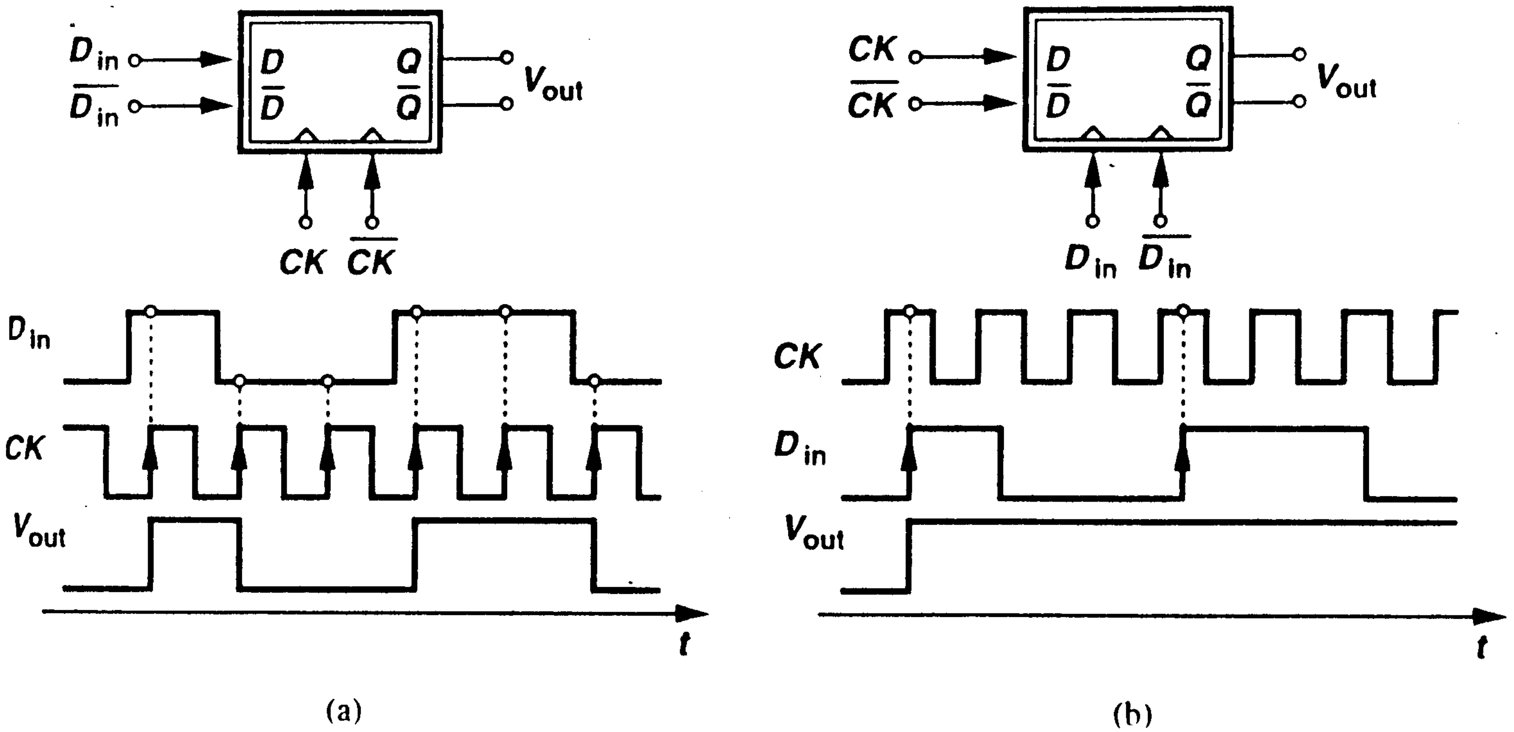
data lags clock → output = 1

data leads clock → output = 0

bang-bang PD.



very high gain



(a)

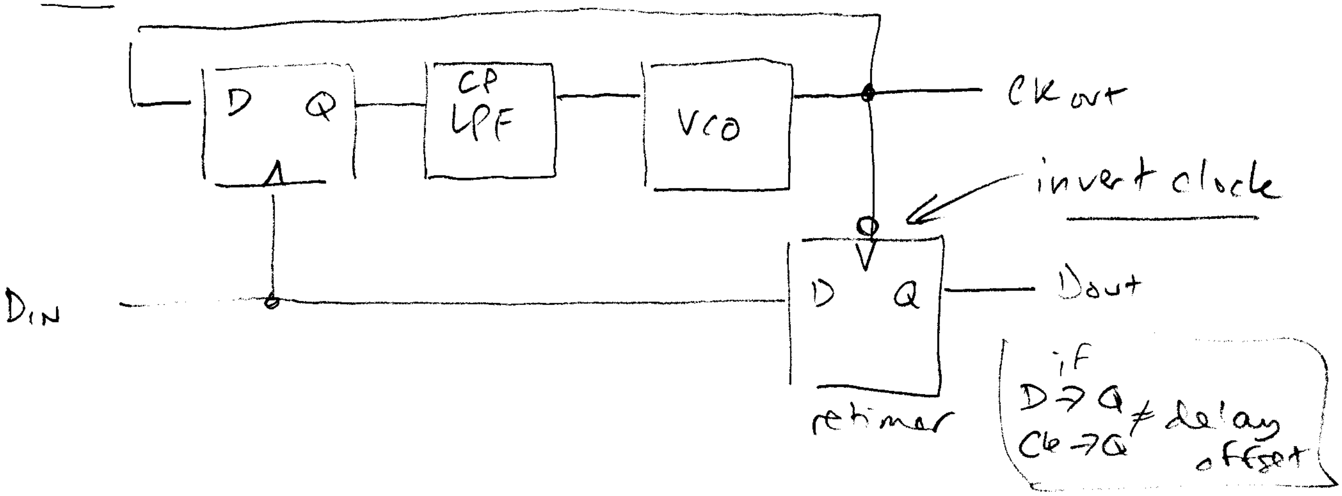
(b)

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

Razavi. 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.

CDR



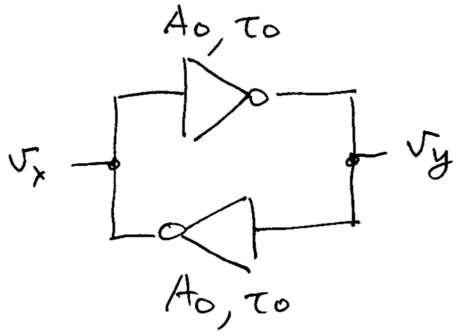
But, as the PD drives the phase error toward zero, the DFF runs into metastability. This limits its effective gain.

Capture. f_{vco} and f_{Data} are different
beat note @ $f_{vco} - f_{Data}$
drives PLL toward lock. (if within LPF BW)

When locked, zero crossings of clock are sampled.

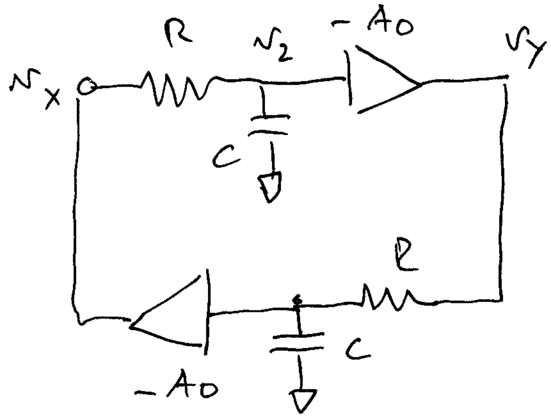
dithering problem with long seq. of 1 or 0

Simple latch model.



$\tau_0 = RC$

represent as single dominant pole for determining transient response.



$$\frac{v_x - v_2}{R} - C\frac{dv_2}{dt} = 0$$

$$-v_2 - \tau_0 \frac{dv_2}{dt} = -v_x$$

$$v_2 = \frac{-v_y}{A_0}$$

$$\frac{v_y}{A_0} + \frac{\tau_0}{A_0}\frac{dv_y}{dt} = -v_x$$

$$v_y + \tau_0 \frac{dv_y}{dt} = -A_0 v_x$$

Likewise,

$$v_x + \tau_0 \frac{dv_x}{dt} = -A_0 v_y$$

subtract

$$\tau_0 \frac{d(v_x - v_y)}{dt} = -(1 - A_0)(v_x - v_y)$$

initial condition $V_{xy0} = (v_x - v_y)|_{t=0}$

$$v_x - v_y = V_{xy0} \exp\left[(A_0 - 1) \frac{t}{\tau_0} \right]$$

growing with time since $A_0 >> 1$

BUT!

with time constant $\tau_0 / (A_0 - 1)$

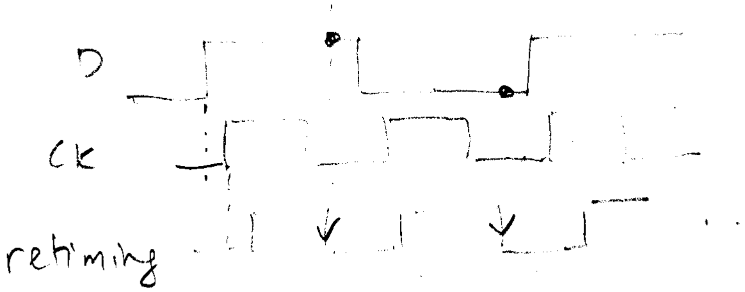
and if V_{xy0} is very small, it would take a long time to reach valid logic levels.

We have skew coming from unequal

Ck → Q and D → Q delay through

PD and retiming FFs.

This can skew the sampling point



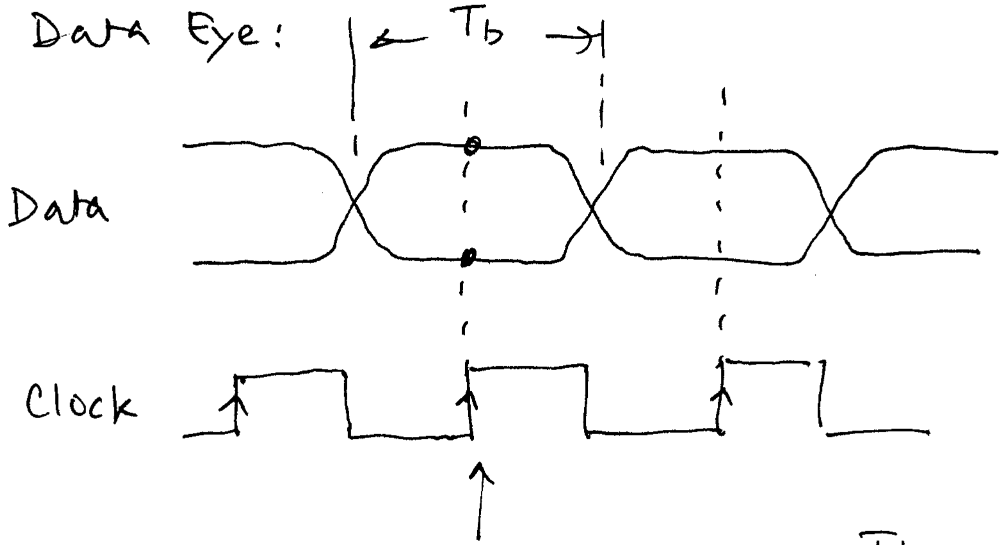
FF kickback (random) can cause VCO jitter

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:



ideal sampling point is at $\frac{T_b}{2}$,

centered between data edges.

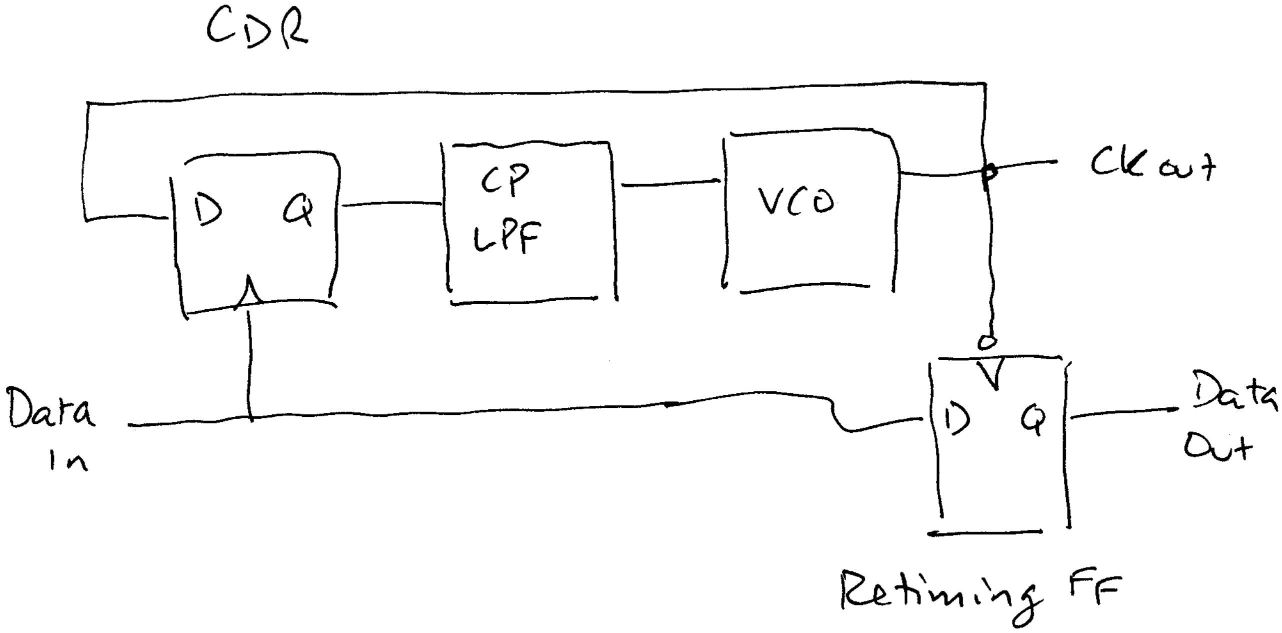
"Phase Margin" = 180° in this case because clock can drift ± 180° for ideal clock and data where rise and fall times << T_b.

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)

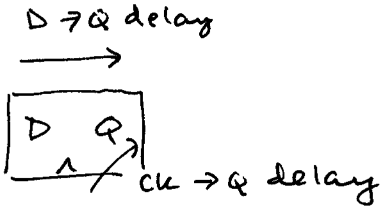
Amplitude noise can also lead to sampling error. A "0" or a "1" can be misinterpreted.

These effects cause the Bit Error Rate to increase.


CDR
D
Q
CP
LPF
VCO
Ck out
Data
In
D
Q
Data
Out
Retiming FF

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.



delays are often different

This leads to a phase error in the PD corresponding to the delay difference ΔT.
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 outright, 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 →

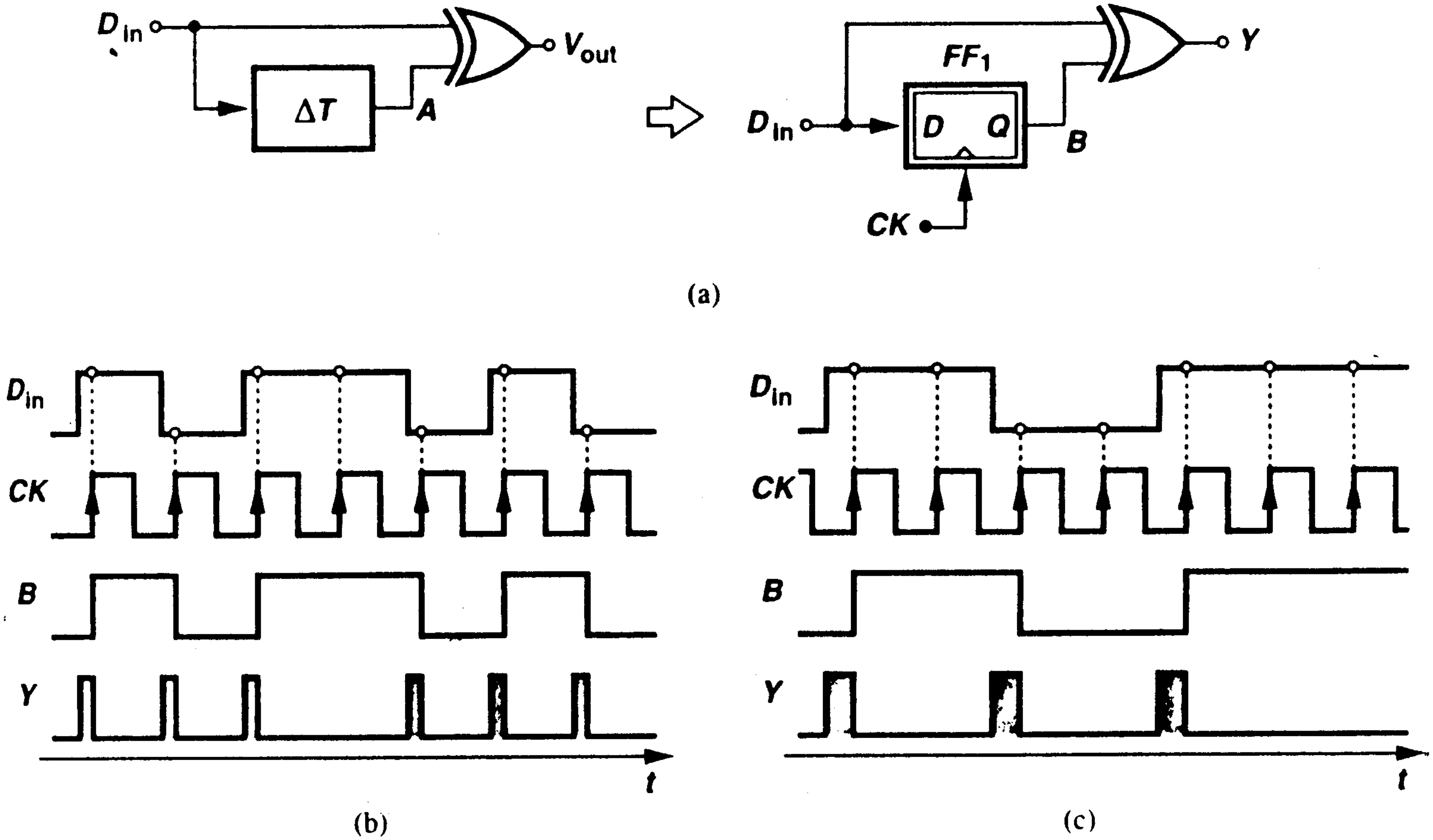


Figure 9.20 (a) Simple PD using synchronous edge detection. (b) PD output for a data pattern. (c) PD output for a different data pattern.

This ambiguity can be resolved if the proportional pulses from Y are ~~also~~ compared against reference pulses with fixed width $T_{ck}/2$. These must also appear only on data edges.

Fig. 9.21 Hogge PD

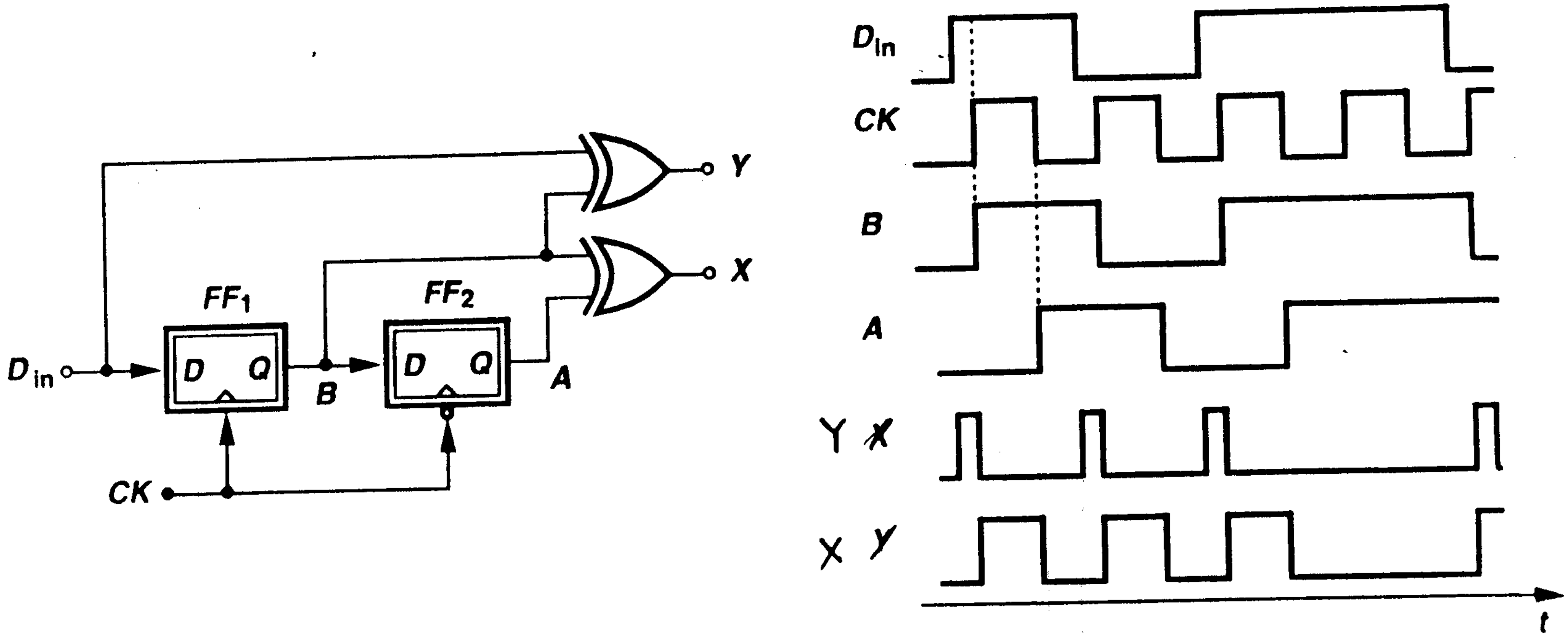
Retimed data can be obtained from either B or A, however, A is shifted by $T_{ck}/2$.

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

1. Ck to Q delay in FF1 widens the pulse at Y by ΔT.

   Ck → Q of FF2 just shifts the ref. pulse, width is the same.

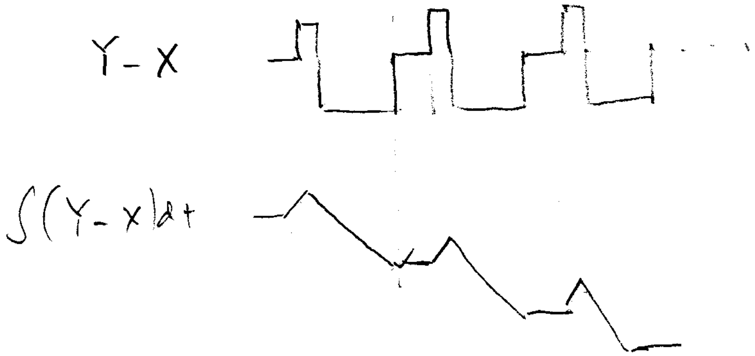
   So, D_{in} and Ck maintain a skew of ΔT. This reduces the phase margin. Especially important for high data rates.



(a)

(b)

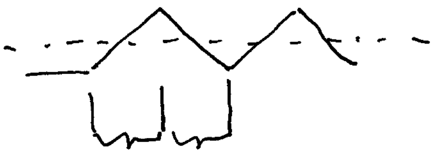
Figure 9.21 (a) Hogge phase detector and (b) its waveforms.



2. Hogge also has a problem with the $T_{ck}/2$ 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.

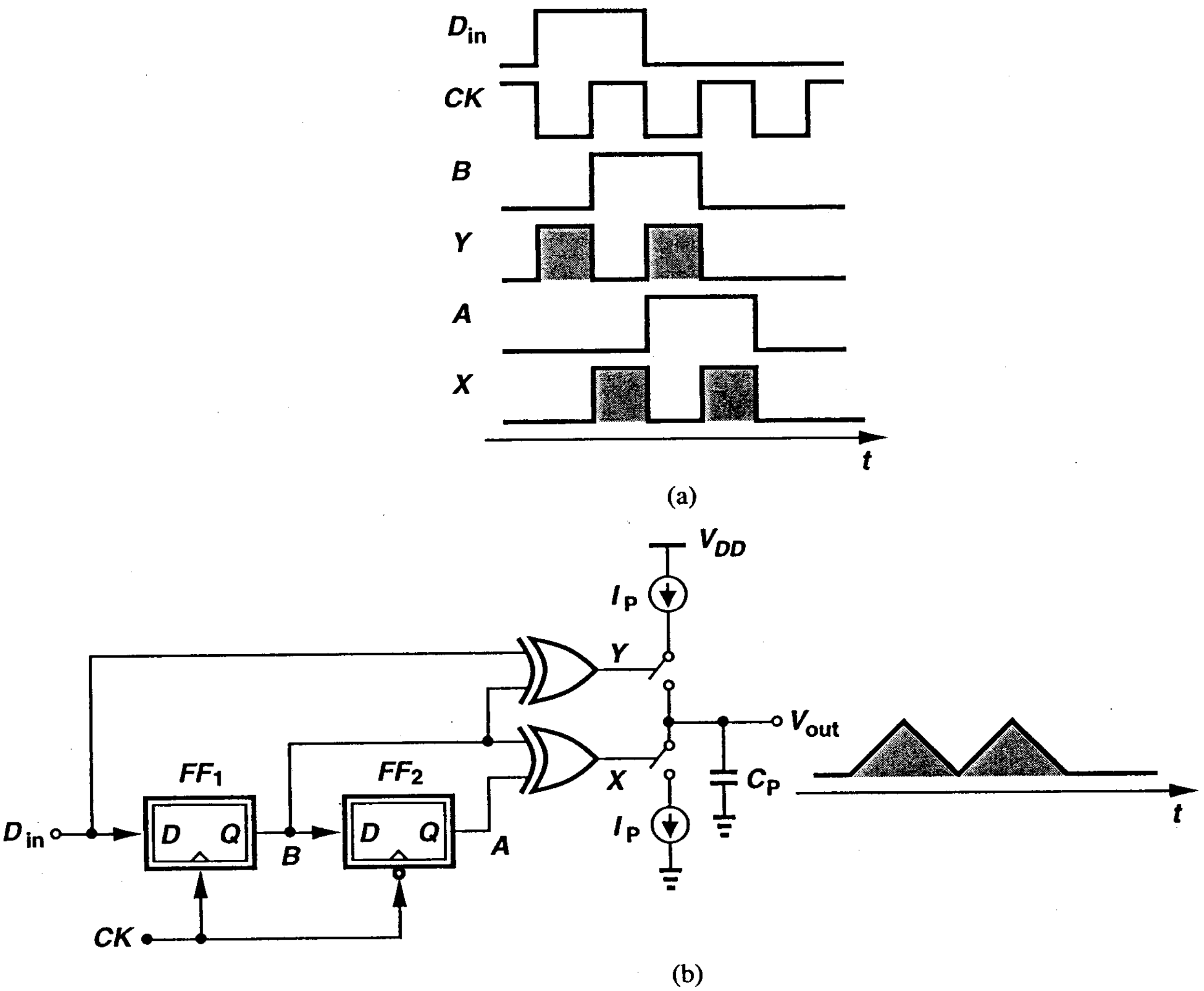


Figure 9.25 (a) Hogge PD waveforms, (b) resulting triwave produced by a charge pump.

B. Razavi, Design of Integrated Circuits for Optical Communications, McGraw-Hill, 2003.

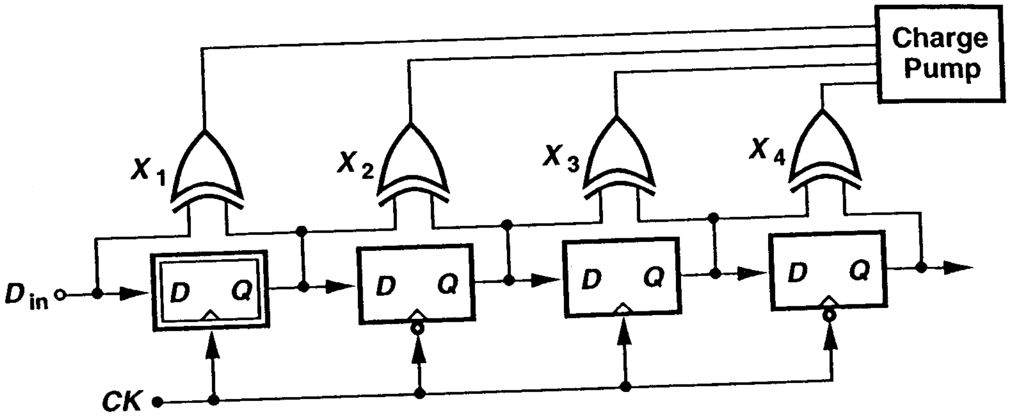


Figure 9.26 Modified Hogge PD to remedy the triwave issue.

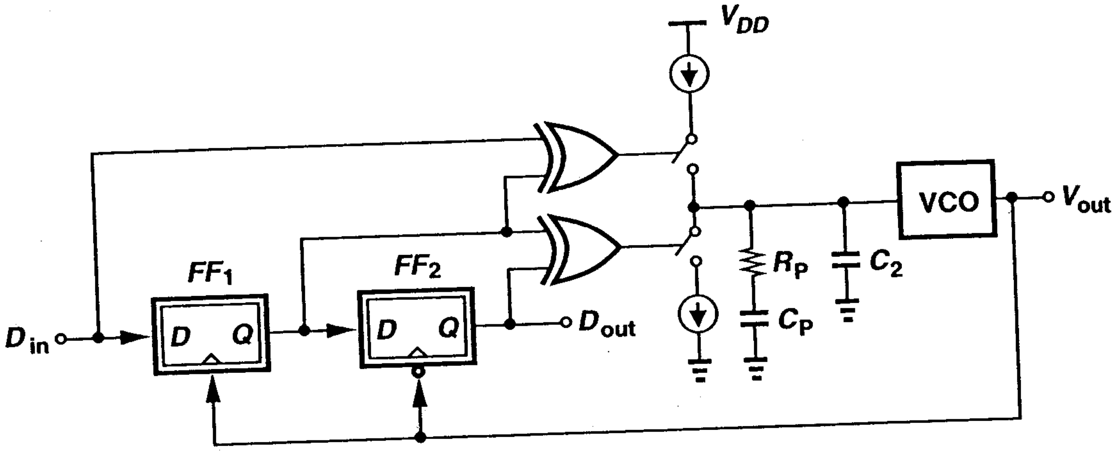


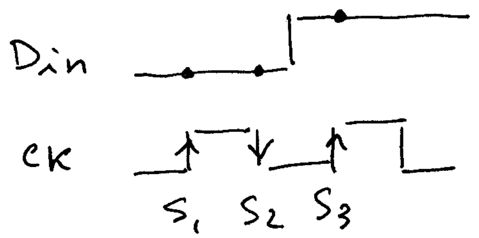
Figure 9.27 CDR circuit using Hogge PD.

B. Razavi, Design of Integrated Circuits for Optical Communications, McGraw-Hill, 2003.

Alexander or Early-Late PD.

Fig 9.28

Multiple points are sampled.



1 1 0
0 0 1 = early

0 1 1 = late
1 0 0

0 0 0 = no transition
1 1 1 (do nothing)

$S_1 \oplus S_2$ and $S_2 \oplus S_3$

1	0	late
0	1	early
0 [illegible]	0 [illegible]	} no data

Fig. 9.29

X and Y outputs drive a charge pump or when converted to current, can be subtracted. Fig 9.31.

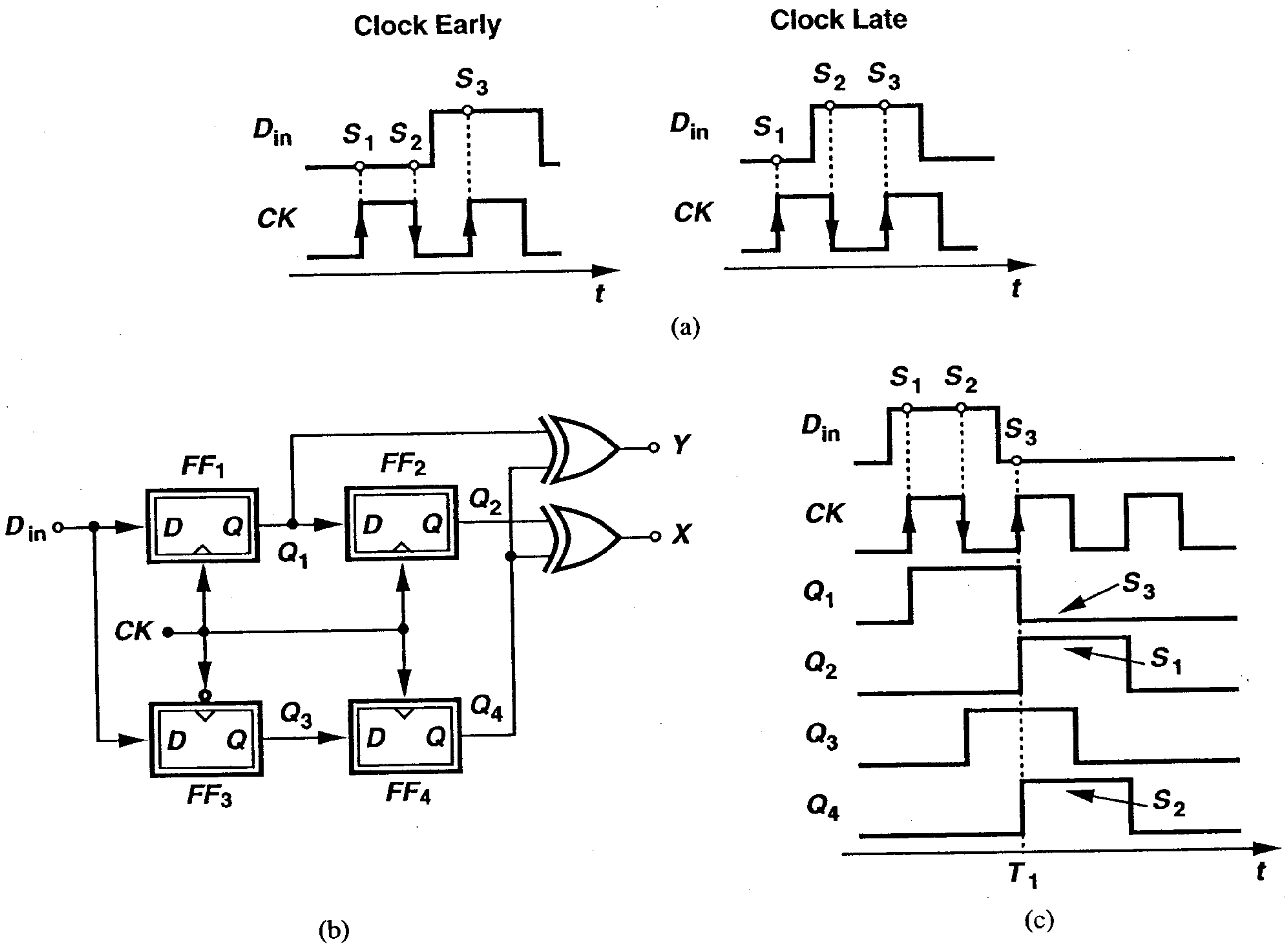


Figure 9.28 (a) Three-point sampling of data by clock, (b) Alexander phase detector, and (c) its waveforms.

B. Razavi, Design of Integrated Circuits for Optical Communications, McGraw-Hill, 2003.

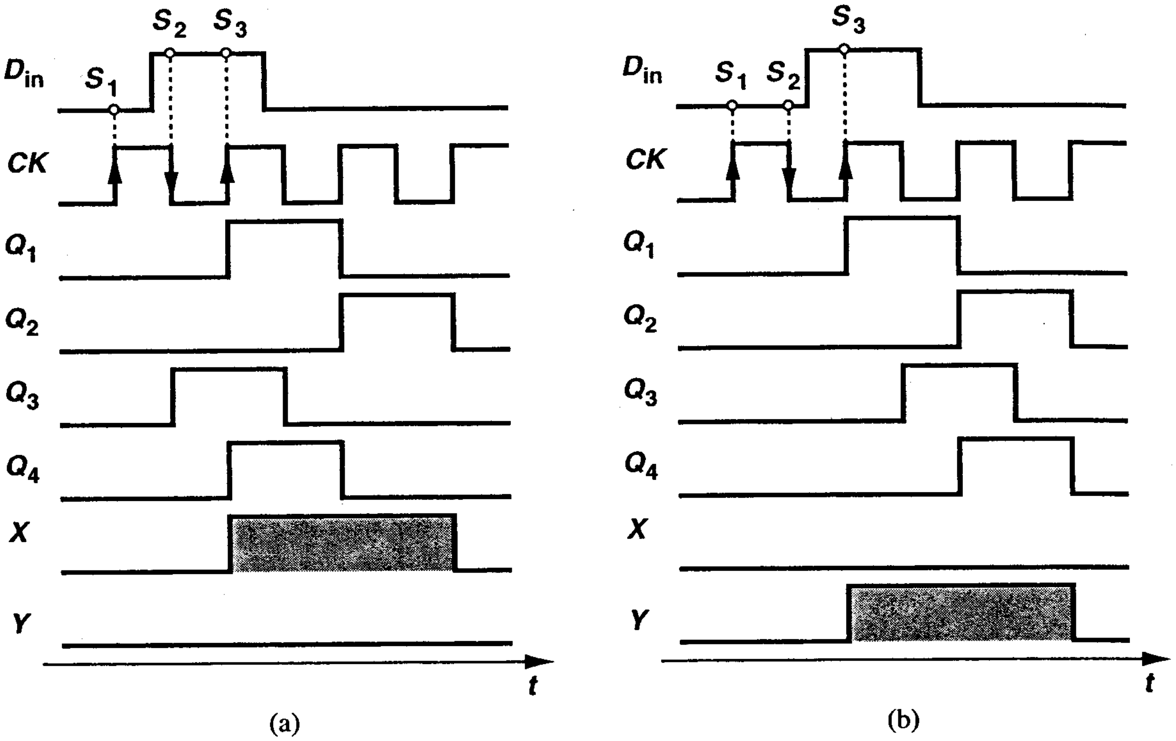


Figure 9.29 Alexander PD waveforms for (a) late and (b) early clock.

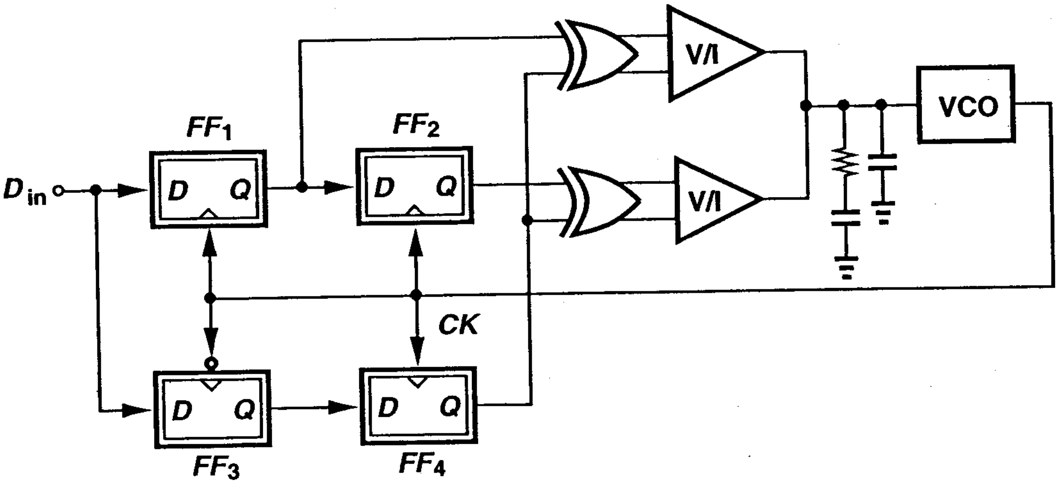


Figure 9.31 CDR circuit using Alexander PD.

B. Razavi, Design of Integrated Circuits for Optical Communications, McGraw-Hill, 2003.

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 accomodate process and temperature variations. If f_{VCO} is too far from f_{DATA}, loop will not lock.

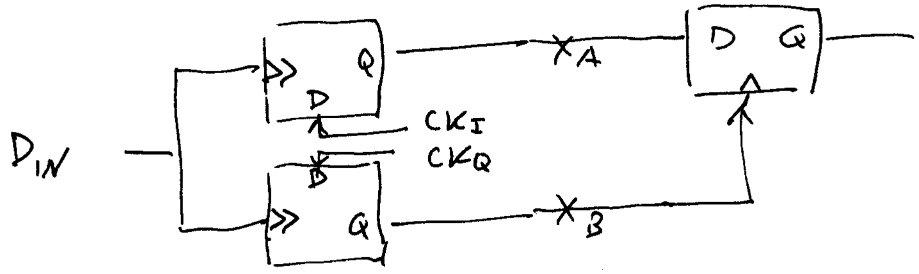
Thus, some assistance is needed to force the VCO to the data frequency.

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 necassary

If data freq. ≠ f_{VCO}, sampling points on CK_I and CK_Q will drift. Transition polarity of X_A and X_B can be used to produce bang-bang FD.

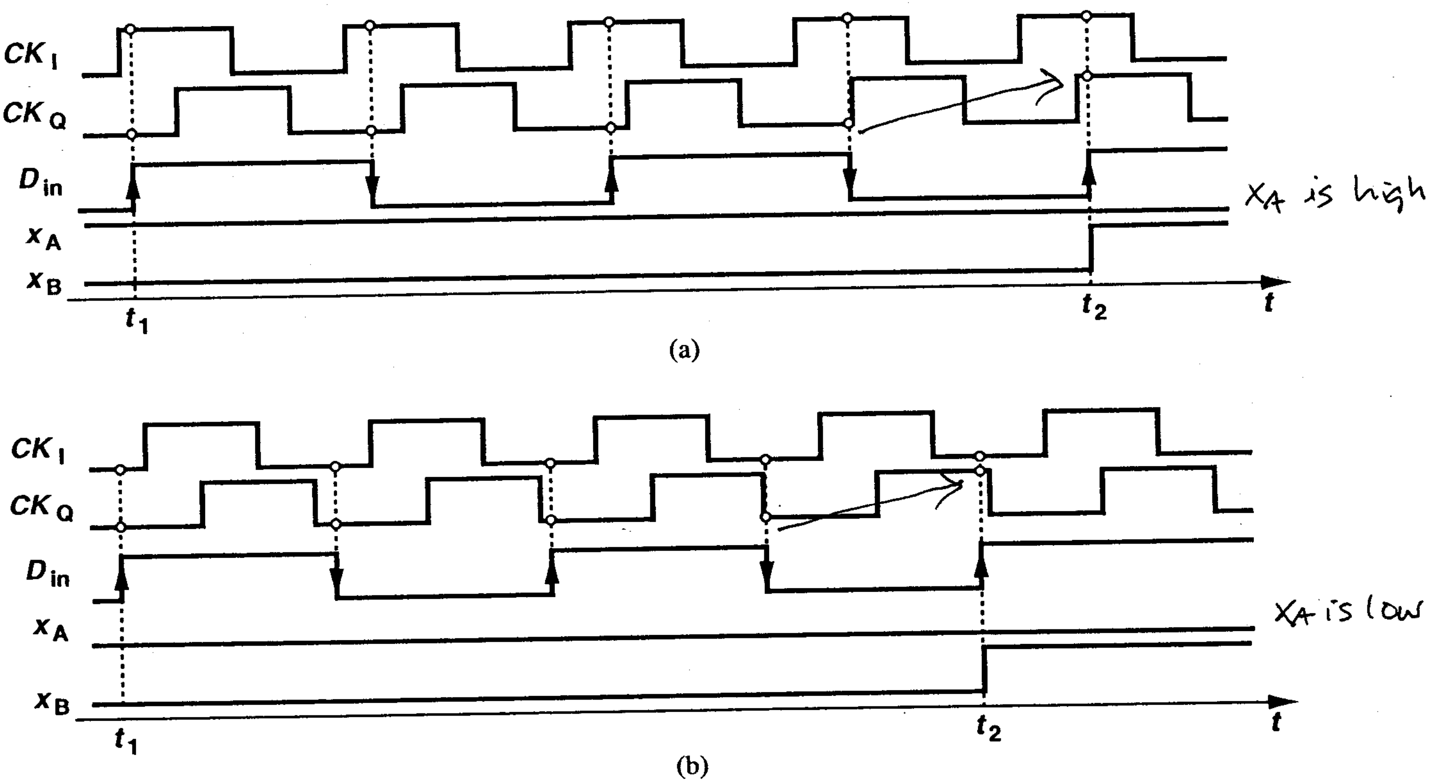


Figure 9.43 FD waveforms for fast and slow clocks.

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

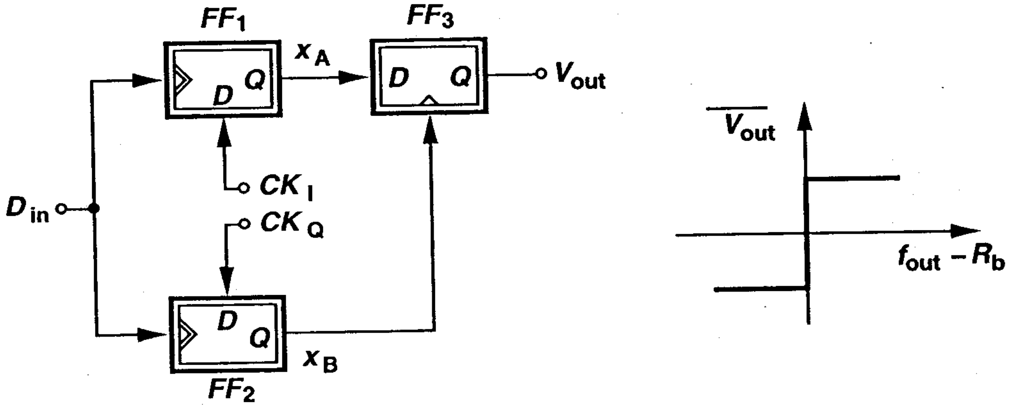


Figure 9.44 Complete frequency detector.

B. Razavi, Design of Integrated Circuits for Optical Communications,
McGraw-Hill, 2003.

In the CDR 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 Kvco may be very high in systems with low V_{DD}, ripple can be reduced by using a fine input for phase control.
- FD 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.

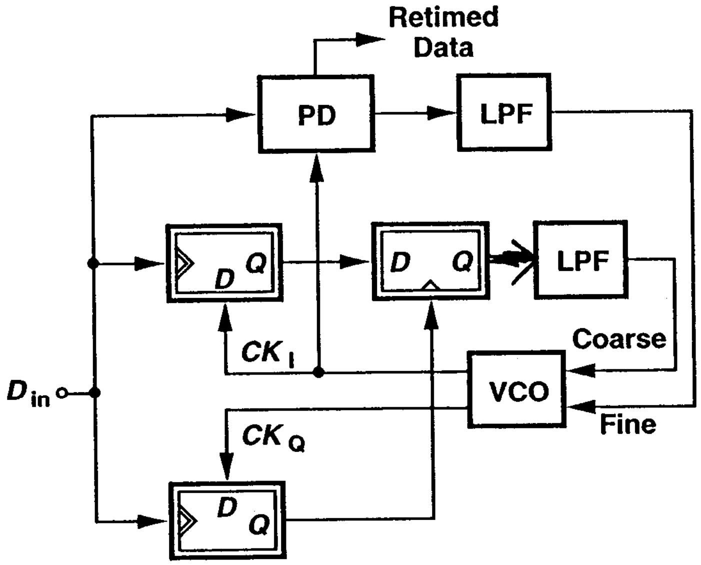


Figure 9.47 CDR architecture with coarse and fine VCO control.

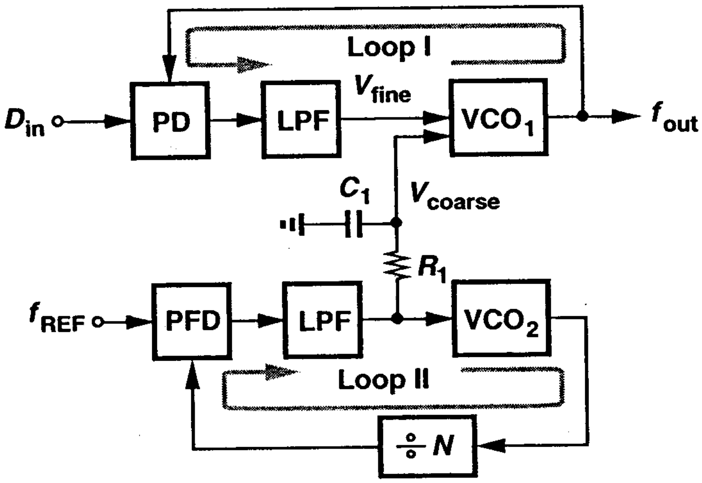


Figure 9.48 CDR architecture using two VCOs.

Razavi op. cit.