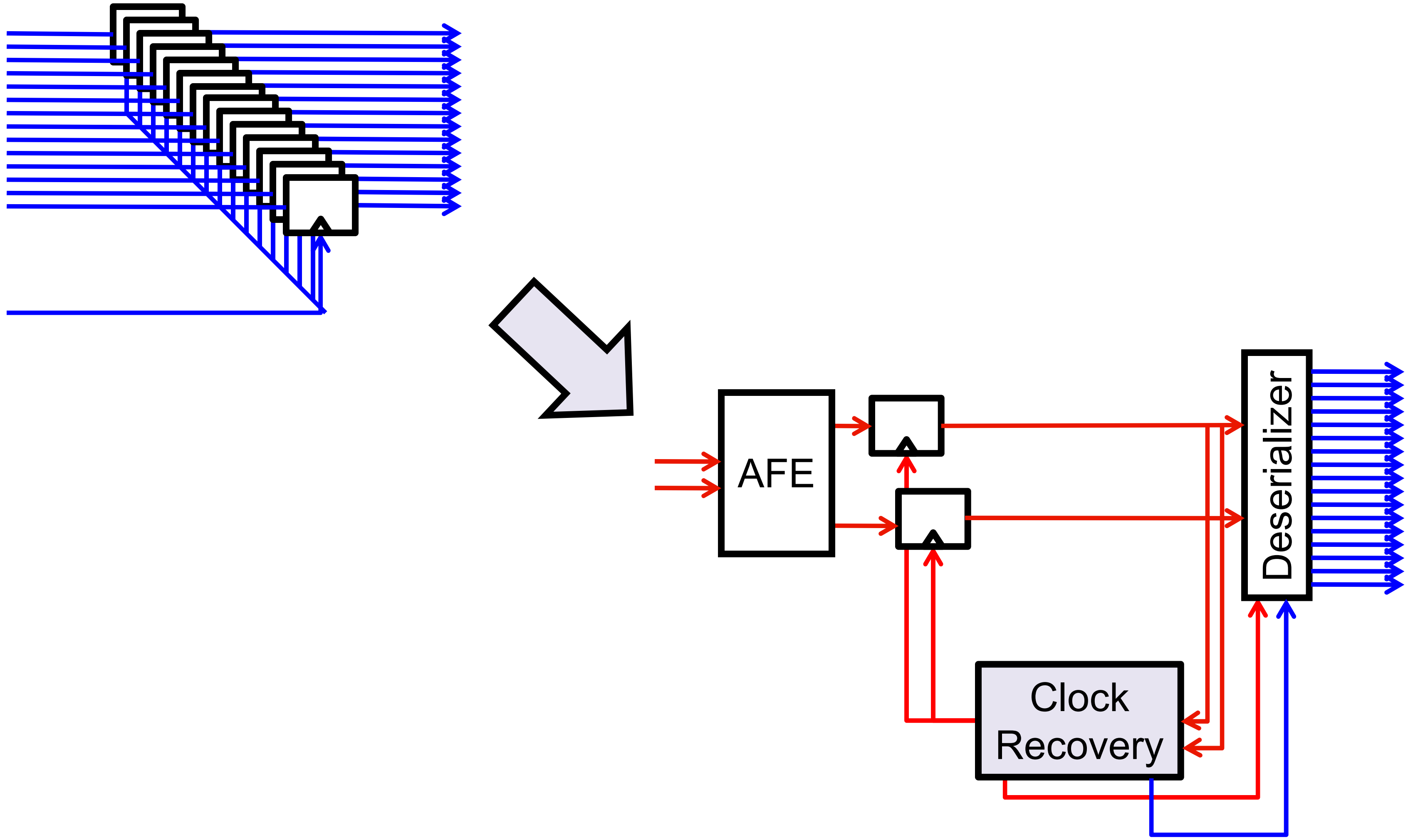
DPLL Based Clock and Data Recovery

John T. Stonick
jts@synopsys.com

Overview

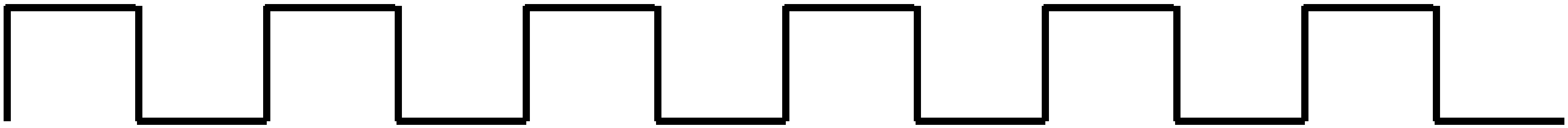
- Focus is on clock recovery
- What is a CDR?
- What is a DPLL based CDR?
- Development of the architecture
- Linear model
- Understanding the operation (making sense of the model)
- Design examples

Parallel vs Serial Links

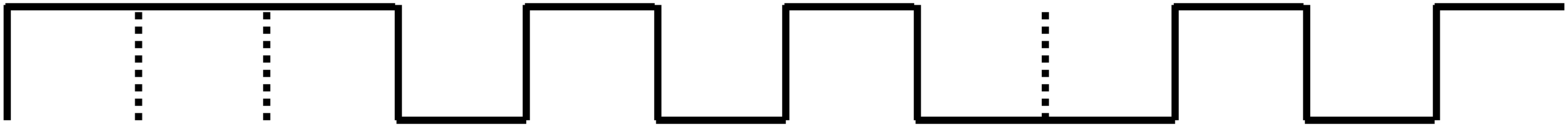


Clock Recovery

Clock has regular occurrences of edges

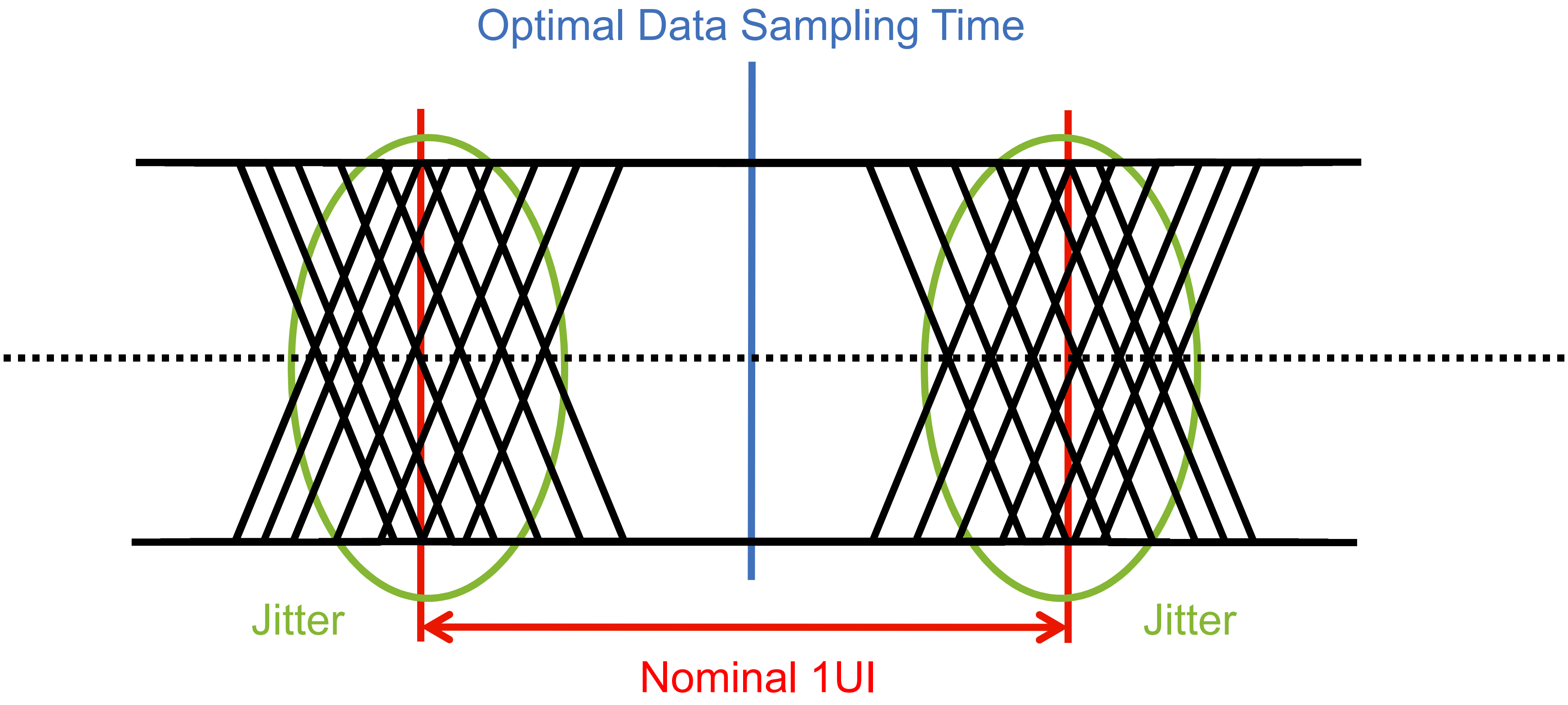


Data has “missing” edges



- Clock recovery from data edges
  - Nyquist data (101010…) looks like a clock at ½ data rate
  - Random data has average edge density is .5
    - May go for long periods with no transitions
- CDR’s job is to generate a sampling clock synchronous with data from non-uniform set of edges

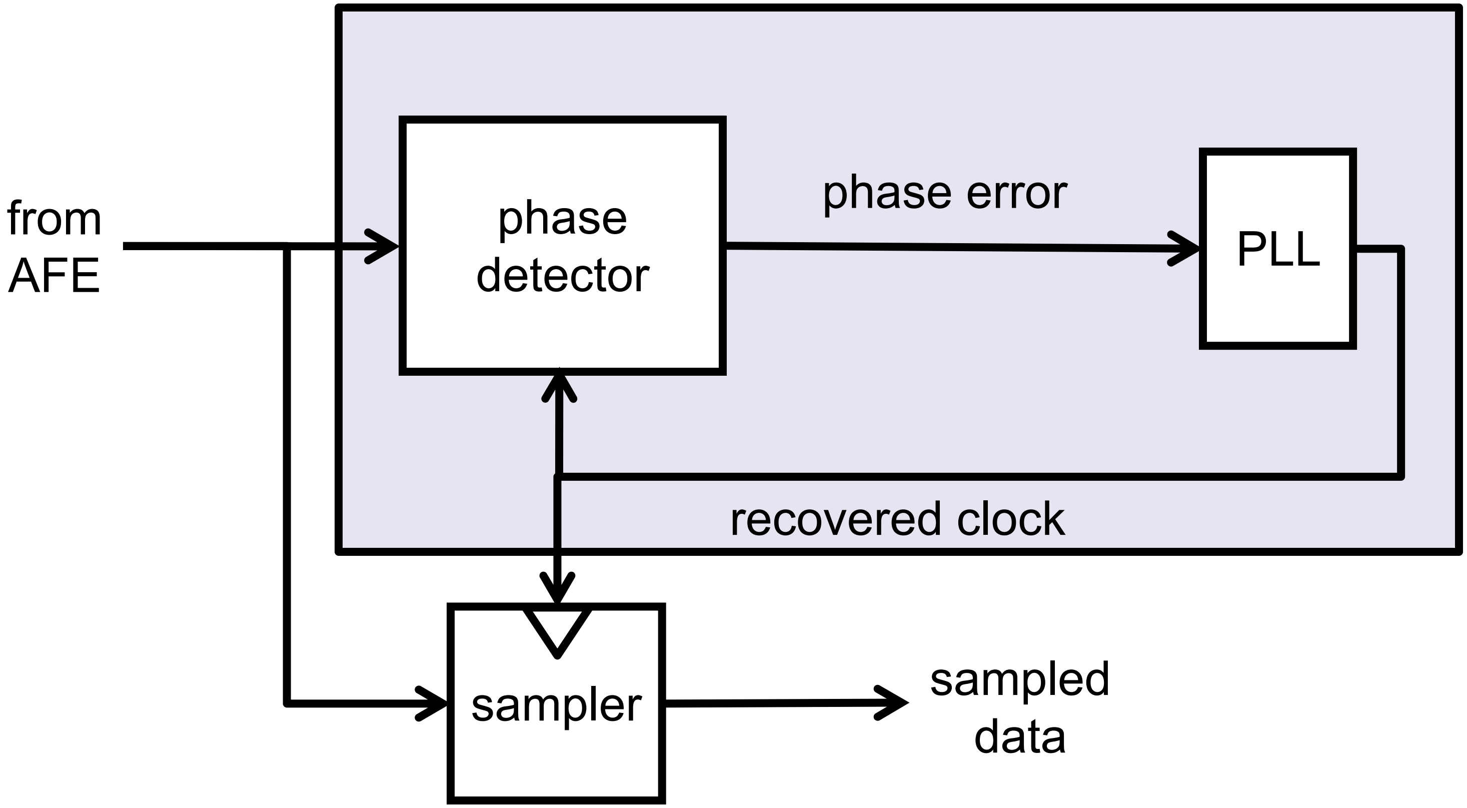
Clock Recovery Basics



CDR Types

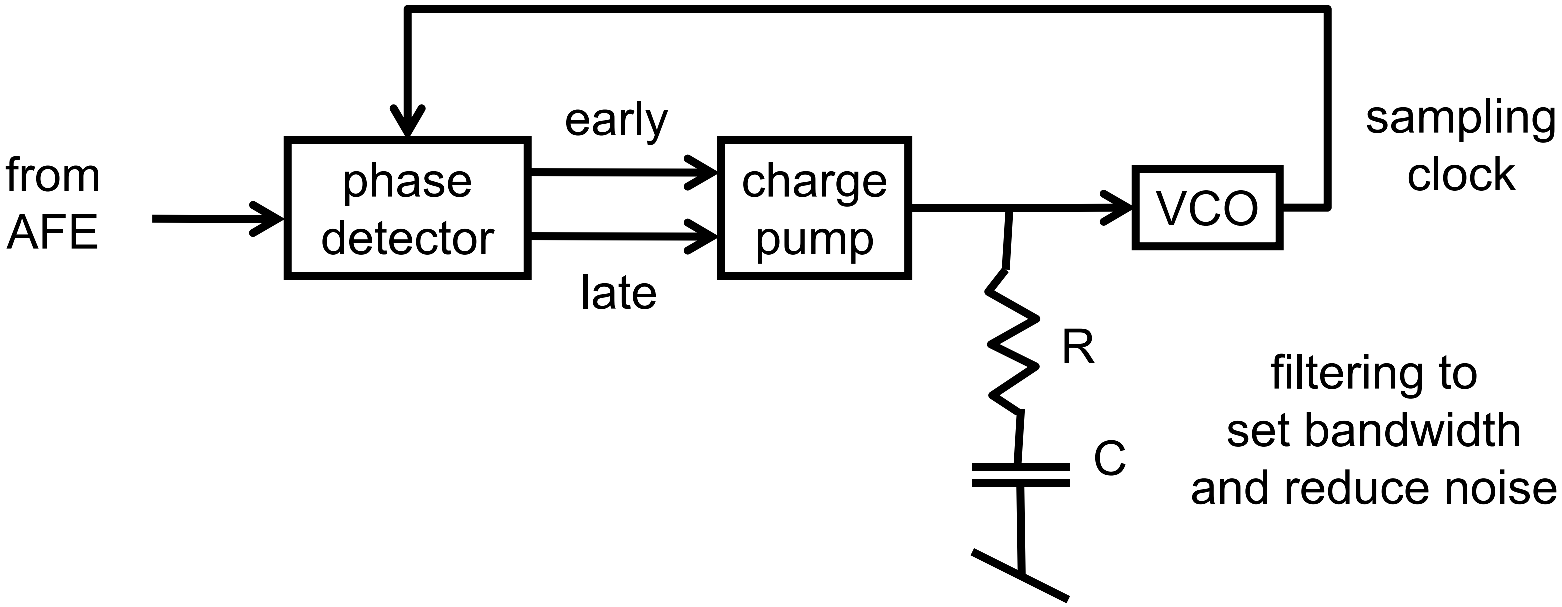
- Analog
  - PLL based with linear phase detector
  - **PLL based with bang-bang phase detector**
- Digital
  - Oversampling
    - Multiple samples per UI
    - Pick the best sample (farthest from transition)
  - DPLL based
    - **Based on a traditional analog bang bang CDR**
    - Replace analog blocks with digital equivalents
    - Saves power and area, reduces PVT variation and improves testability with respect to analog implementation

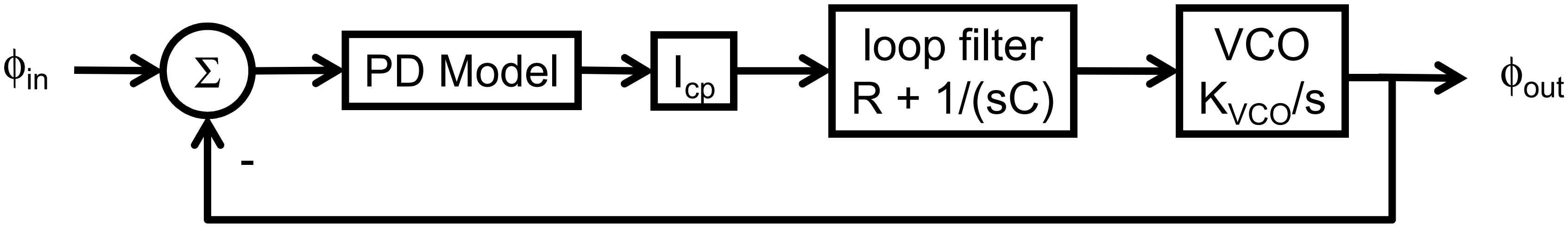
Analog PLL-Based CDR Block Diagram



Force recovered clock to be in quadrature with incoming data

Analog PLL-Based CDR Detail

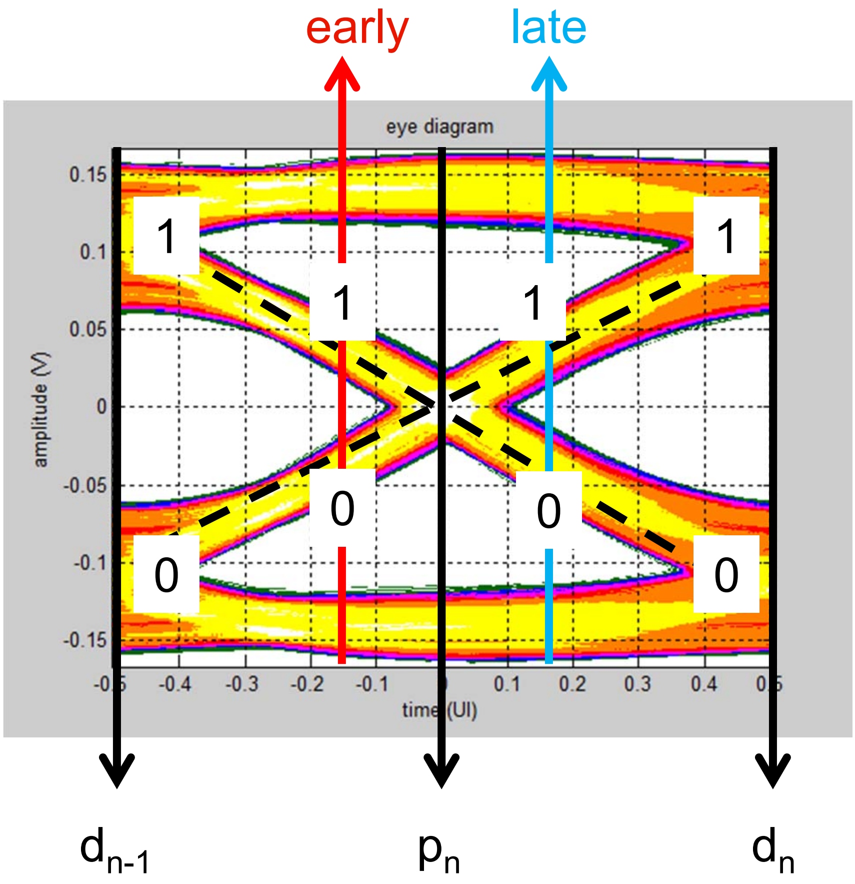




Phase Detector Types

- Linear (Hogge)
  - Produces a late **and** an early pulse on each transition
  - Late and early pulses are **variable** width. *Difference* in pulse widths *proportional* to phase error
  - Lock condition is late and early pulses of equal width on each clock cycle
  - Difficult to implement without static phase error and only useful in analog implementations
- Bang bang (Alexander)
  - Produces a late **or** an early pulse on each transition
  - Late and early pulses always the **same** width
  - Lock condition is equal average number of early and late pulses (time average to 0)
  - Inherently less problematic than linear PD, usable in DPLL

Digital View of the Bang-Bang Phase Detector

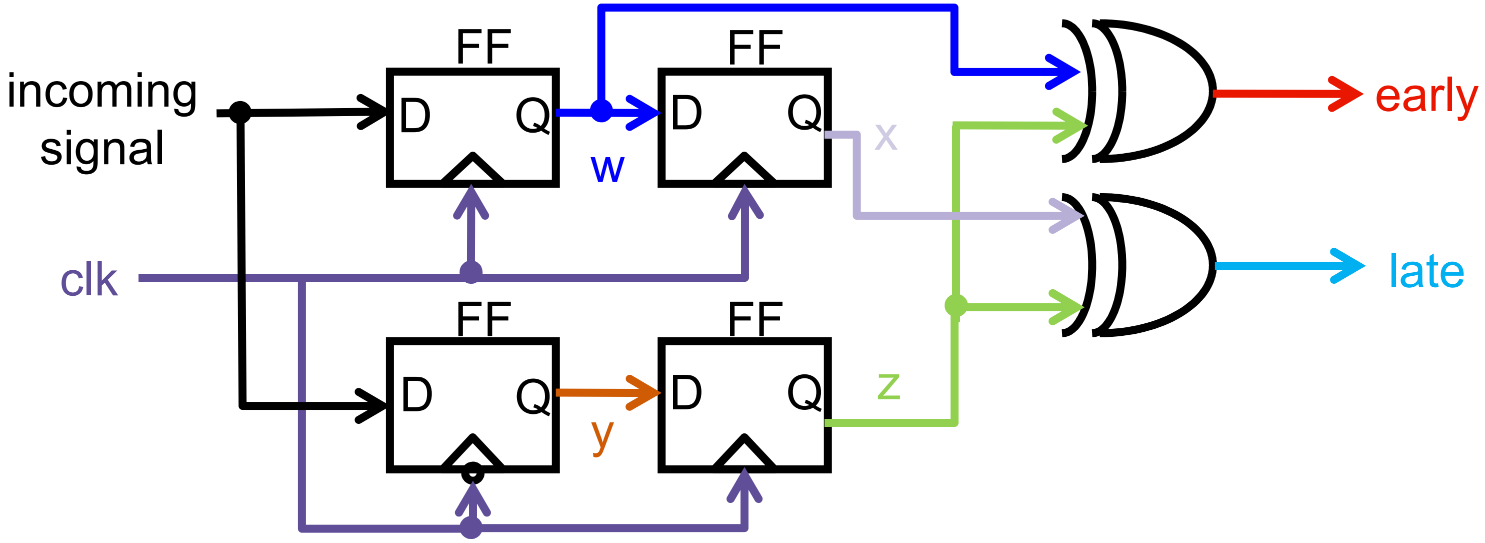


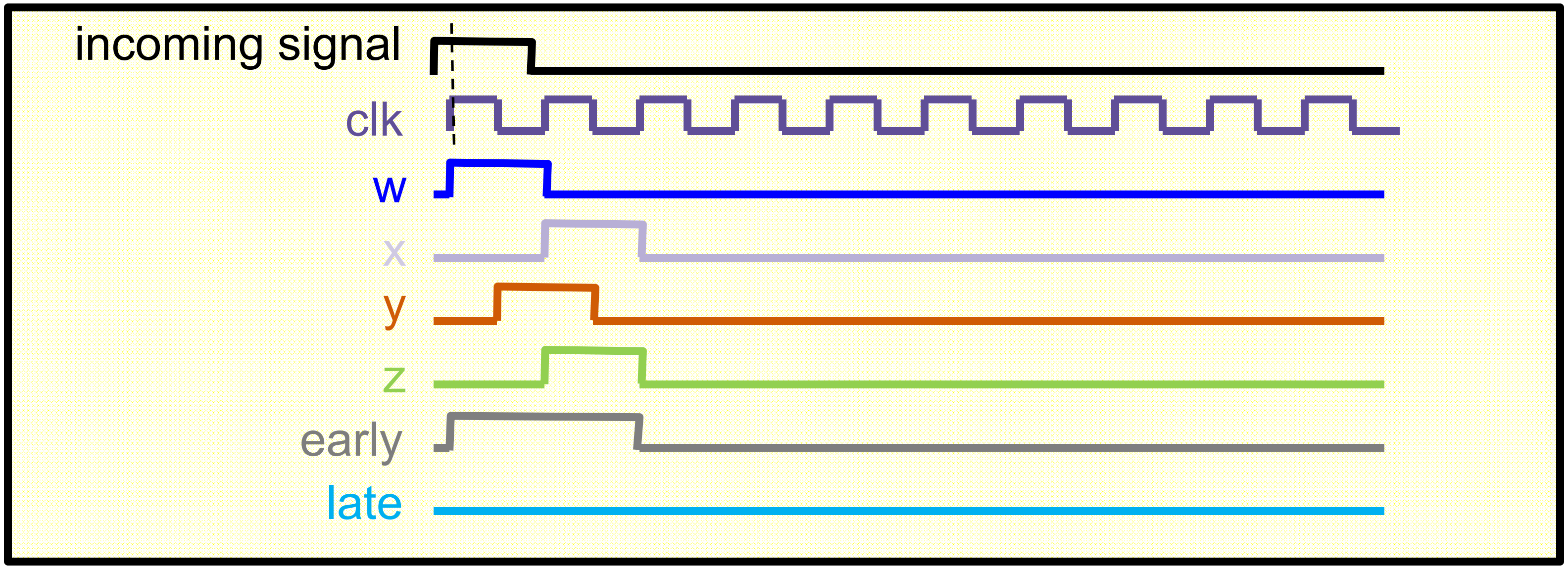
look up table

d_{n-1}	p_n	d_n	phe
0	1	1	late
0	0	1	early
1	1	0	early
1	0	0	late
0	x	0	no trans
1	x	1	no trans

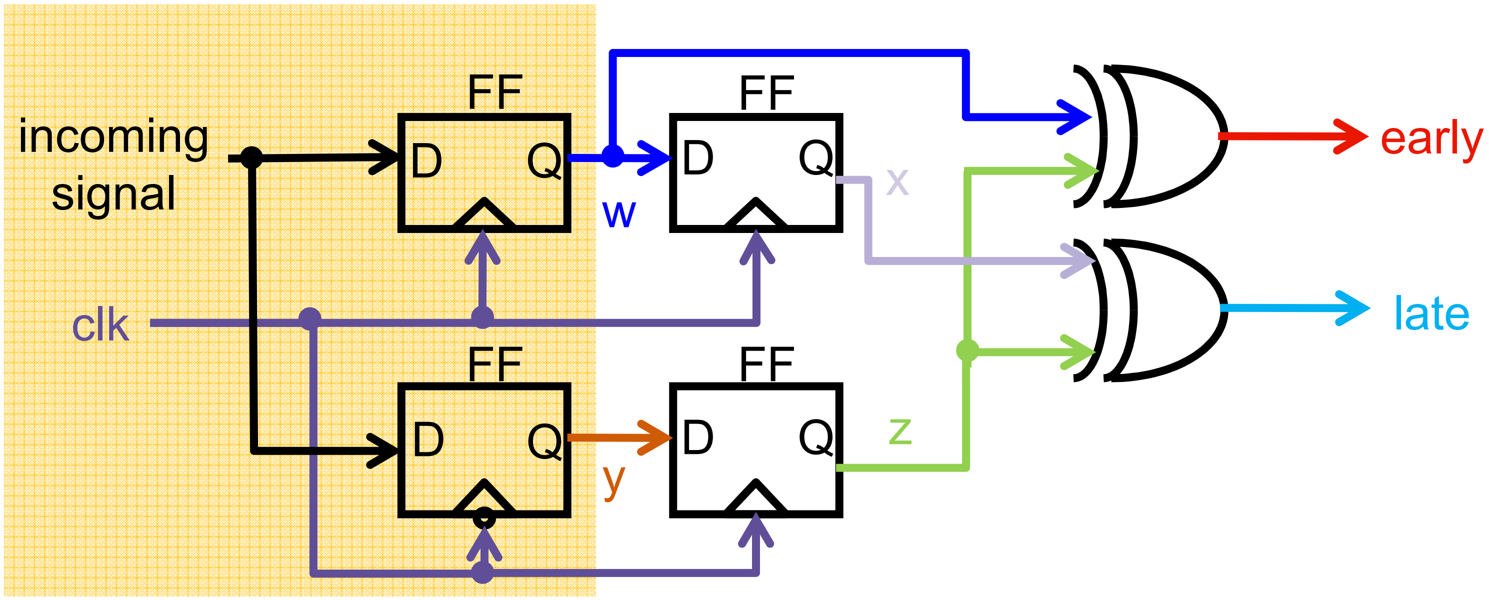
- Output is three levels: (2 bits: early, late, no transition)
- We can take data and phase samples and make phe decisions later

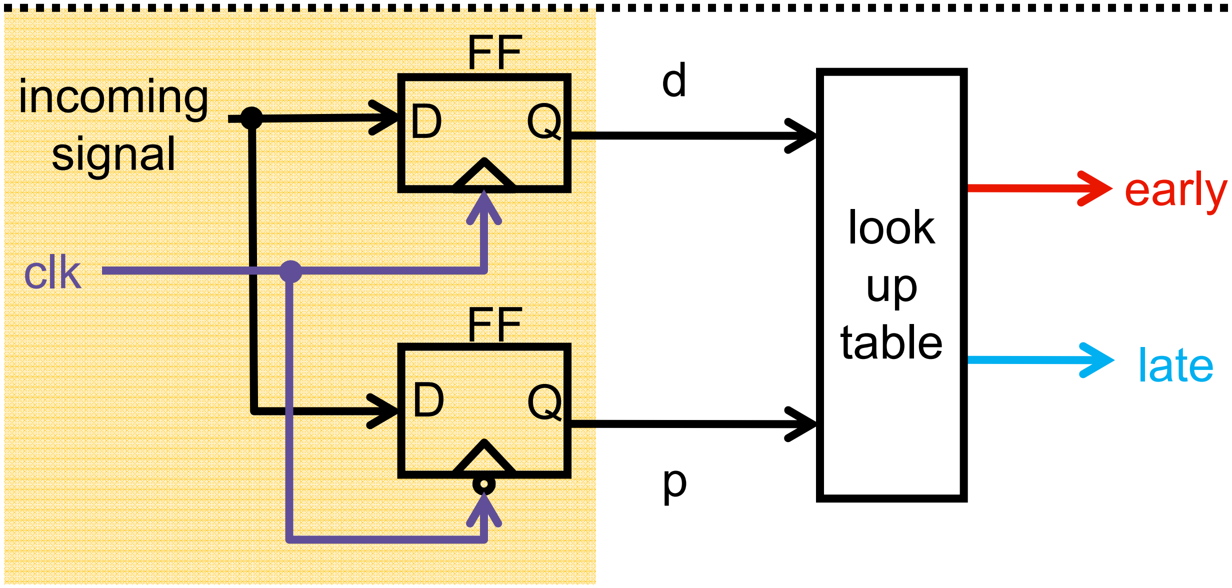
Bang-Bang (Alexander) Phase Detector (Clock is Early)



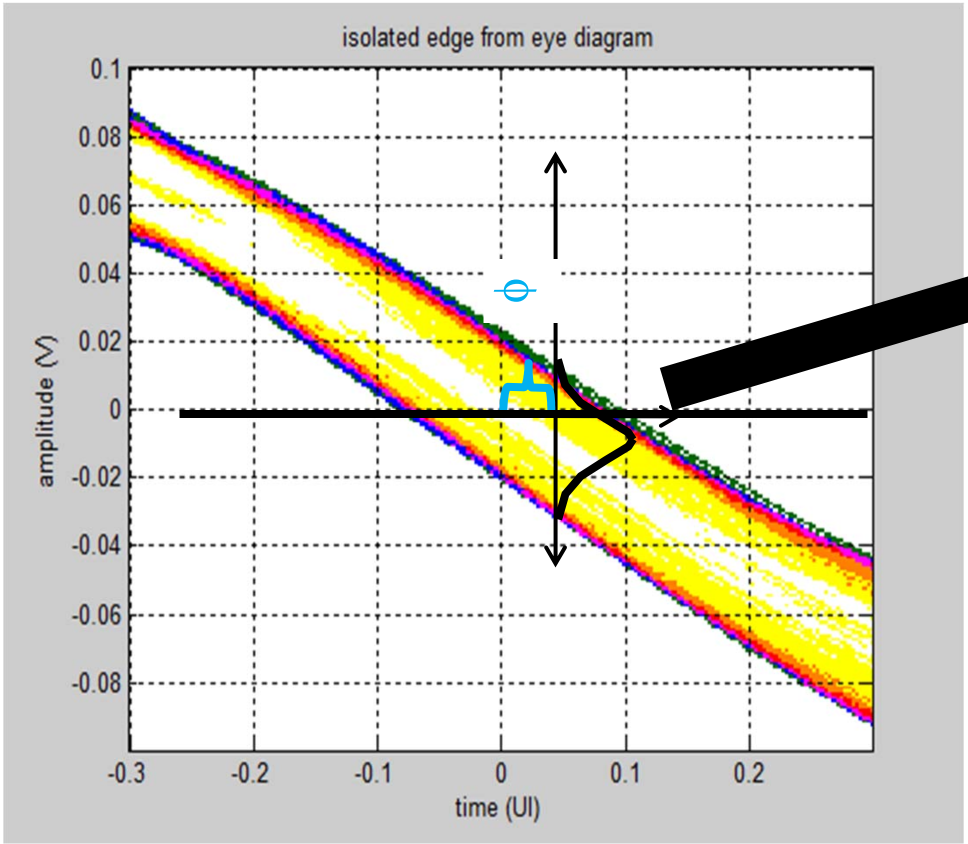


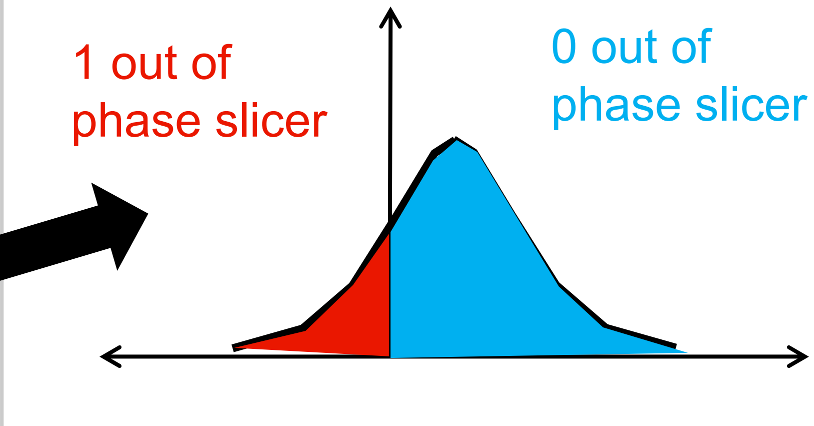
Comparison of Bang-Bang Realizations





Linearizing the Bang-Bang Phase Detector





1 out of phase slicer

0 out of phase slicer

Early probability is area of red
Late probability is area of blue

Time average value coming out of BB PD (μ) with respect to phase offset related to gain!

Arbitrarily assign -1 to early and +1 to late

Linearizing the Bang-Bang Phase Detector

- μ = (1)Pr(late|ϕ) + (-1)*Pr(early|ϕ)
- Each phase offset (ϕ) will have a different mean (μ)
  - BB Gain is slope of average BB output, μ, versus phase offset, ϕ

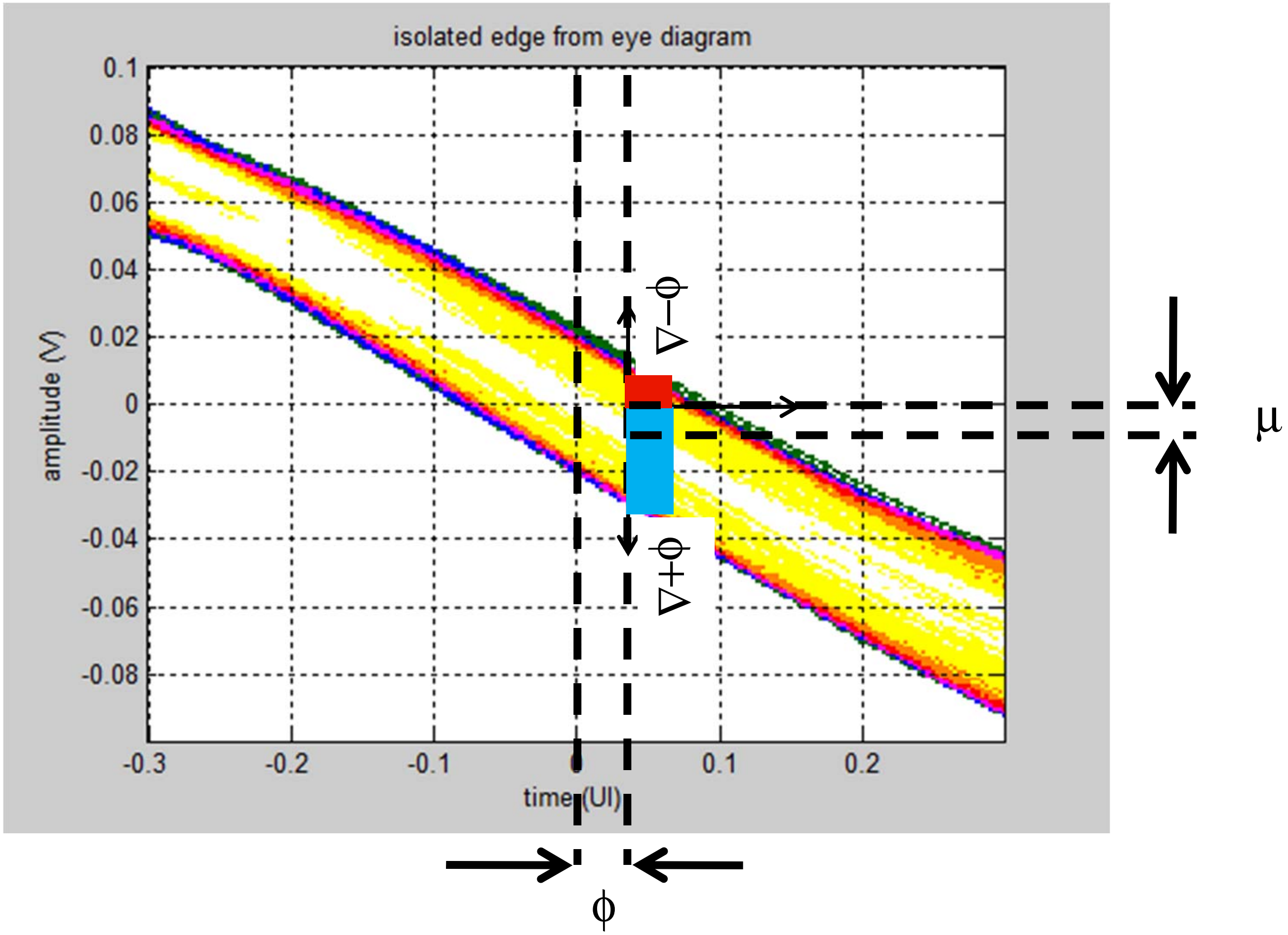
$$\frac{\partial \mu}{\partial \phi}$$

- BB only produces output for a transition and this de-rates the gain. Transition density = .5 for random data

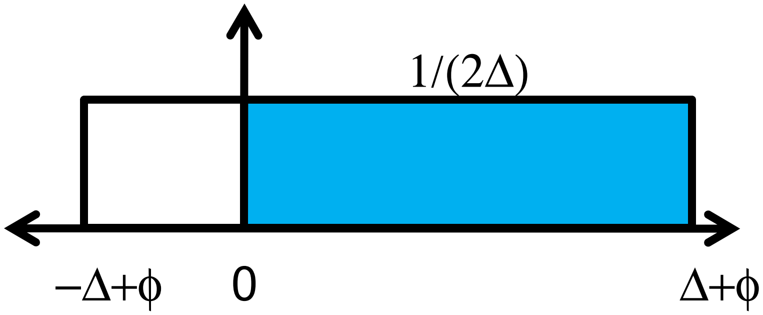
$$K_{BB} = \frac{1}{2}\frac{\partial \mu}{\partial \phi}$$

- Need to assume a PDF for jitter to compute K_{BB}
  - Gaussian (RJ dominated) and uniform (DJ dominated)

Uniform PDF (DJ Dominated)



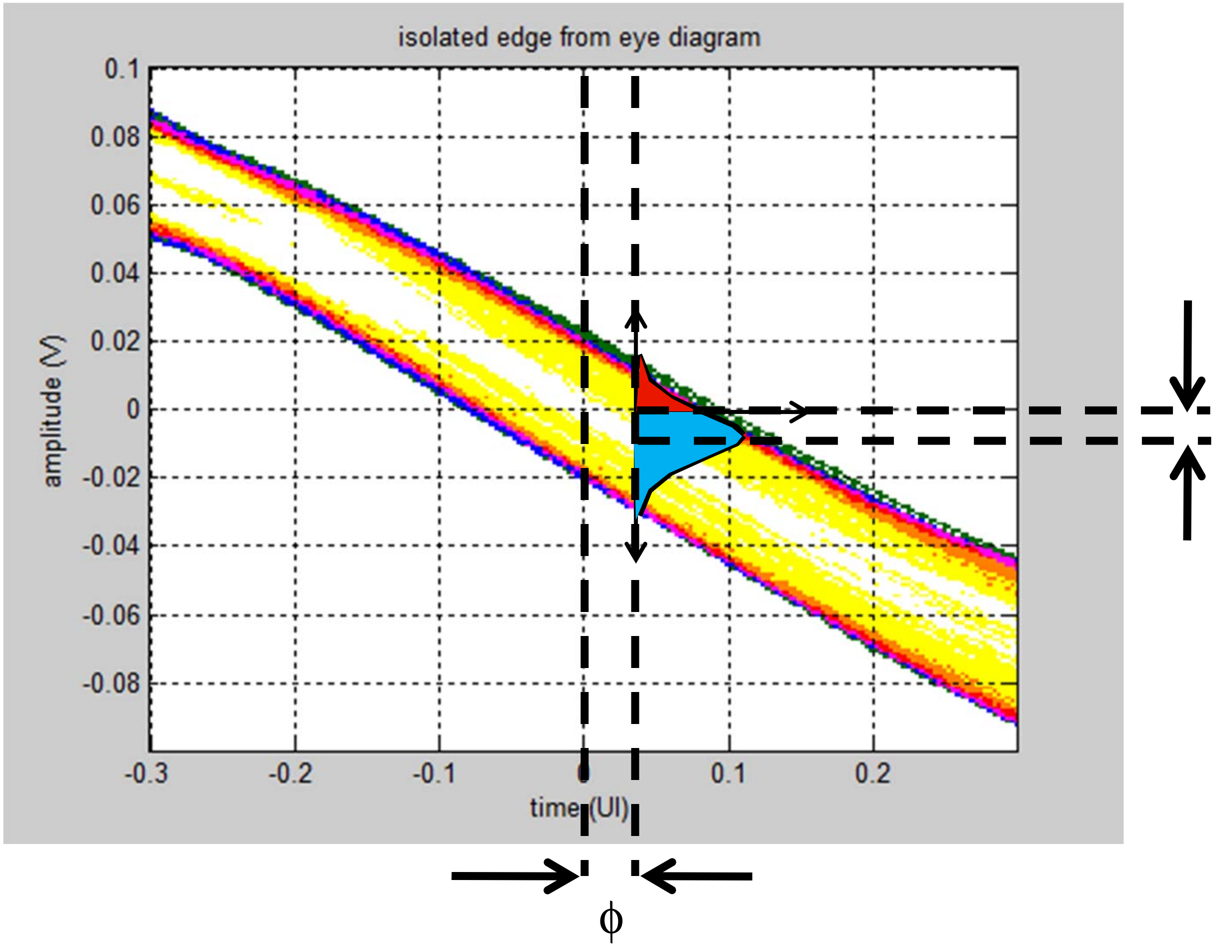
Pr(late|ϕ) for Uniform PDF



$$\sigma^2 = \frac{1}{2\Delta}\int_{-\Delta+\phi}^{\Delta+\phi} x^2 dx - \phi^2 = \frac{\Delta^2}{3} \quad \Rightarrow \quad \Delta = \frac{\sigma}{\sqrt{3}}$$

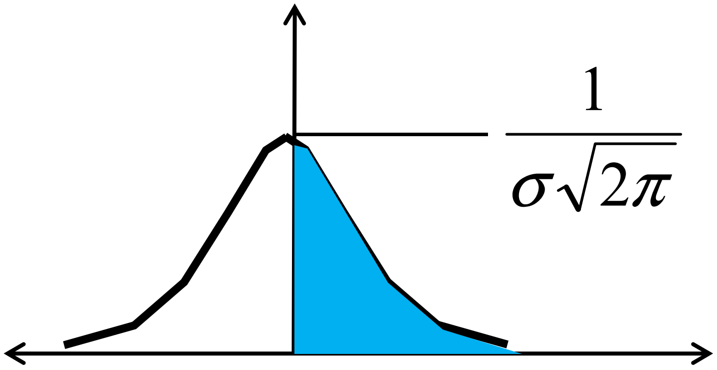
$$\Pr\left(\text{late}\middle|\phi\right) = \frac{1}{2\Delta}\left[\Delta - (-\phi)\right] = \frac{\sigma\sqrt{3}+\phi}{2\sigma\sqrt{3}} = \frac{1}{2}\left(1 + \frac{\phi}{\sigma\sqrt{3}}\right)$$

Gaussian PDF (RJ Dominated)



μ: average of early and late

Pr(late|ϕ) for Gaussian PDF



$$\frac{1}{\sigma\sqrt{2\pi}}$$

mean is -ϕ
standard deviation is σ

$$\Pr\left(\text{late}\middle|\phi\right)=\frac{1}{\sigma\sqrt{2\pi}}\int_{0}^{\infty}e^{\frac{-(x+\phi)^2}{2\sigma^2}}dx$$

$$\Pr\left(\text{late}\middle|\phi\right)=\frac{1}{\sqrt{2\pi}}\int_{\frac{\phi}{\sigma}}^{\infty}e^{\frac{-y^2}{2}}dy$$

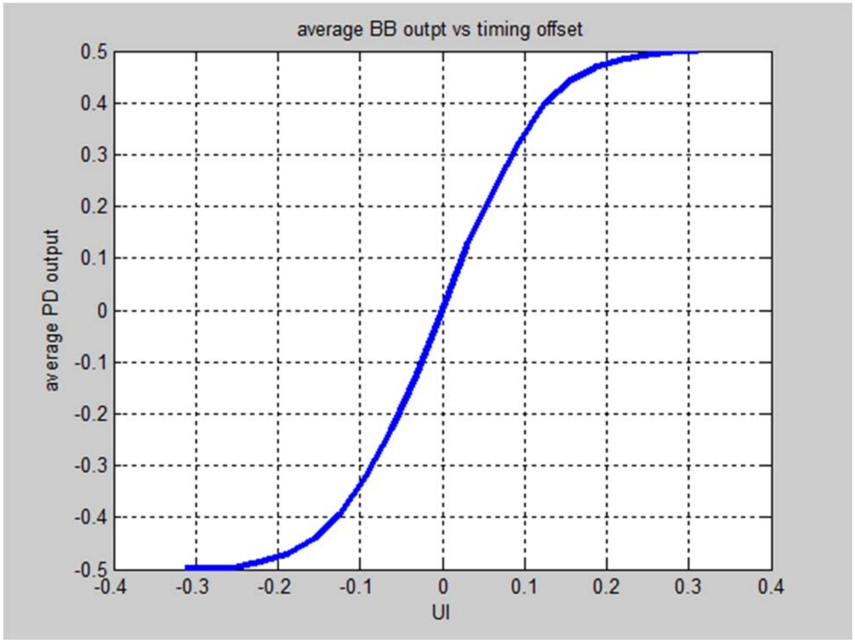
$$Q(A)=\frac{1}{\sqrt{2\pi}}\int_{A}^{\infty}e^{\frac{-y^2}{2}}dy \longrightarrow \Pr\left(\text{late}\middle|\phi\right)=Q\left(\frac{\phi}{\sigma}\right)\approx\frac{1}{2}-\frac{\phi}{\sigma\sqrt{2\pi}}$$

Gain for Bang-Bang Phase Detector

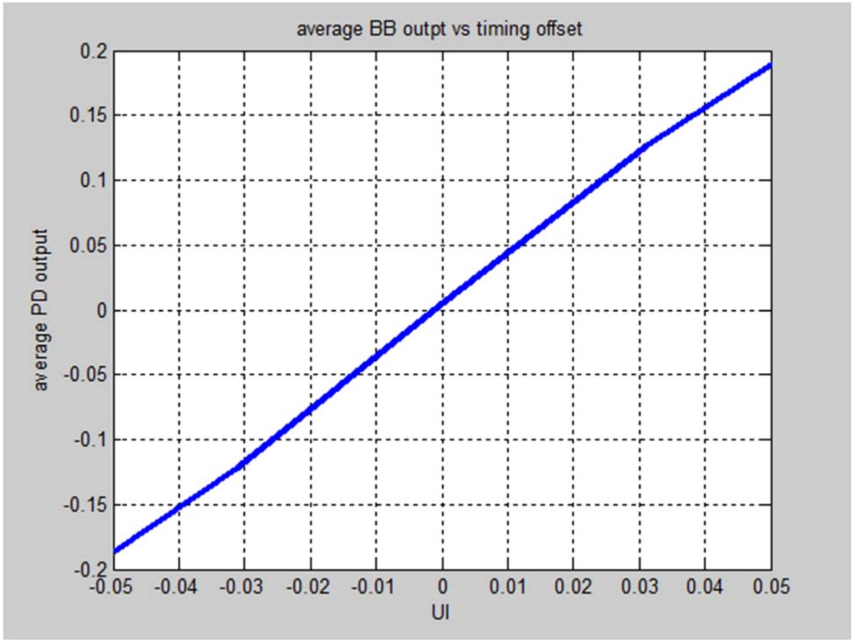
	Uniform	Gaussian
Pr(late|ϕ)	$\frac{1}{2}\left(1+\frac{\phi}{\sigma\sqrt{3}}\right)$	$\frac{1}{2}-\frac{\phi}{\sigma\sqrt{2\pi}}$
μ = 2*Pr(late|ϕ) - 1	$\frac{\phi}{\sigma\sqrt{3}}$	$\frac{2\phi}{\sigma\sqrt{2\pi}}$
$K_{BB}=\frac{1}{2}\frac{\partial\mu}{\partial\phi}$	$\frac{1}{\sigma\sqrt{12}}$	$\frac{1}{\sigma\sqrt{2\pi}}$

BB gain will be somewhere between the two extremes

Verification for Gaussian Jitter, σ = .1UI



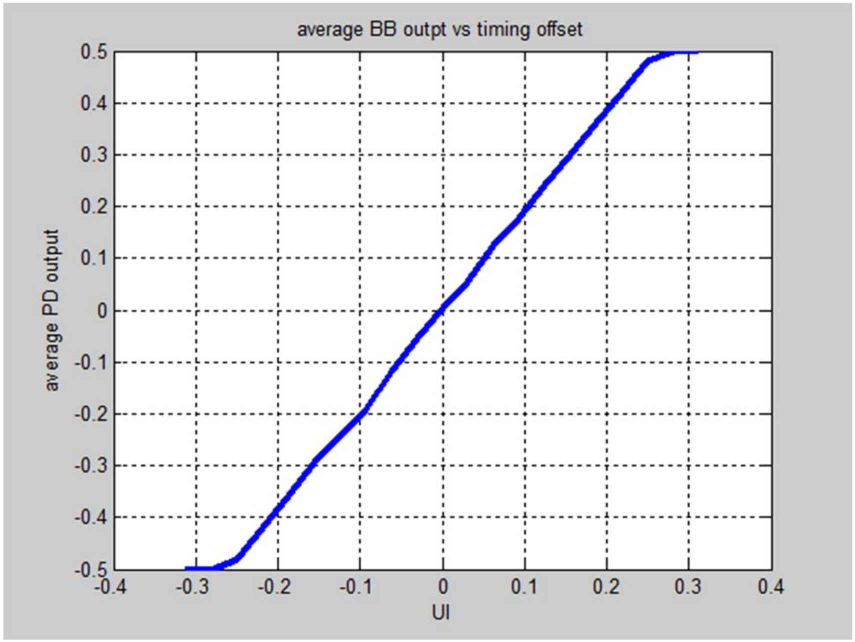
Zoom in on linear region

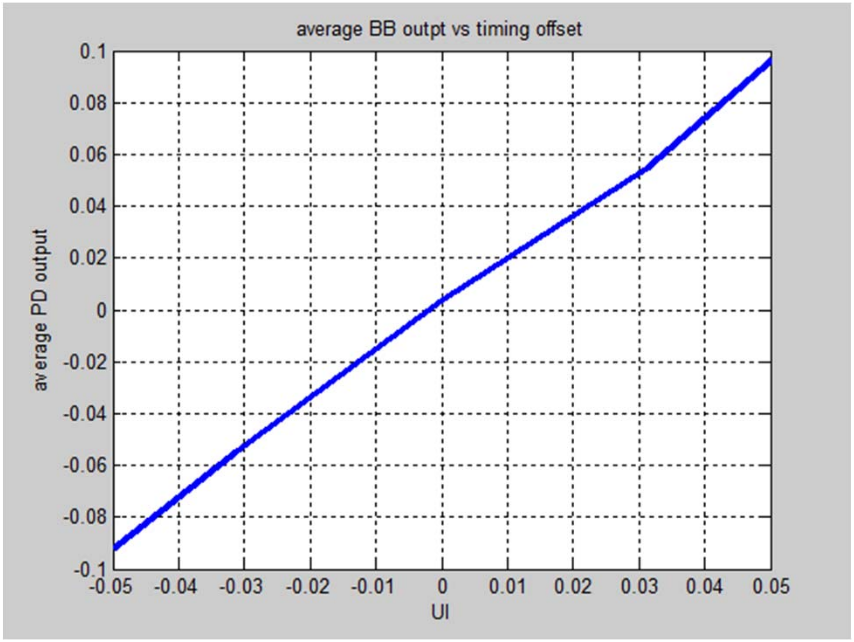


Measured slope: 3.98 calculation: $\frac{1}{.1\sqrt{2\pi}} = 3.99$

Verification for Uniform Jitter, σ = .15UI

Zoom in on linear region





Measured slope: 1.89   calculation: $\frac{1}{.15\sqrt{12}} = 1.92$

Bang-Bang Self-Noise Term (σ_{BB})

- One price we pay for BB PD versus linear PD is the self-noise term
- For small phase errors BB output noise is the full magnitude of the sliced data
  - Average power is the probability of a transition
  - Average transition density is .5, Pr(trans)=.5
  - $\sigma_{\text{BB}}^2 = \text{Pr(trans)} = .5$
- To understand impact we can reflect noise back to the input

$$\sigma_{\text{BBin}}^2 = \frac{\sigma_{\text{BB}}^2}{\text{K}_{\text{BB}}^2} = \frac{.5}{\text{K}_{\text{BB}}^2}$$

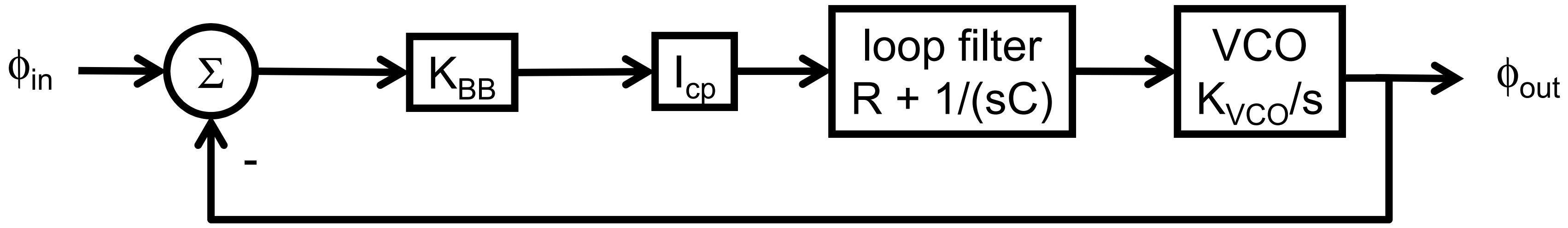
BBPD Self Noise Input Referred

- Lets consider our 2 PDFs
  - Uniform, $\mathrm{K}_{\mathrm{BB}} = \frac{1}{\sigma\sqrt{12}}, \quad \sigma^2_{\mathrm{BBin}} = \frac{.5}{\mathrm{K}^2_{\mathrm{PD}}} = 6\sigma^2, \quad \sigma_{\mathrm{BBin}} = \sqrt{6}\sigma$
  - Gaussian, $\mathrm{K}_{\mathrm{BB}} = \frac{1}{\sigma\sqrt{2\pi}}, \quad \sigma^2_{\mathrm{BB}} = \frac{.5}{\mathrm{K}^2_{\mathrm{PD}}} = \pi\sigma^2, \quad \sigma_{\mathrm{BB}} = \sqrt{\pi}\sigma$
- Input referred jitter from BB PD is proportional to incoming jitter. Assuming they add in power, the increase in jitter is computed to be:
  - Uniform: $\sigma^2 \Rightarrow \sigma^2(1+6), \quad \sigma \Rightarrow \sigma\sqrt{7}$
  - Gaussian: $\sigma^2 \Rightarrow \sigma^2(1+\pi), \quad \sigma \Rightarrow \sigma\sqrt{1+\pi}$

BBPD Model Summary

- BBPD modeled as a gain plus an additive jitter term
- BBPD gain inversely proportional to incoming jitter
  - More jitter leads to lower gain
  - Clean signals provide highest gain
- BBPD noise reflected to input is proportional to incoming jitter and models the jitter increase versus a linear phase detector

Analog BB CDR Small Signal Model



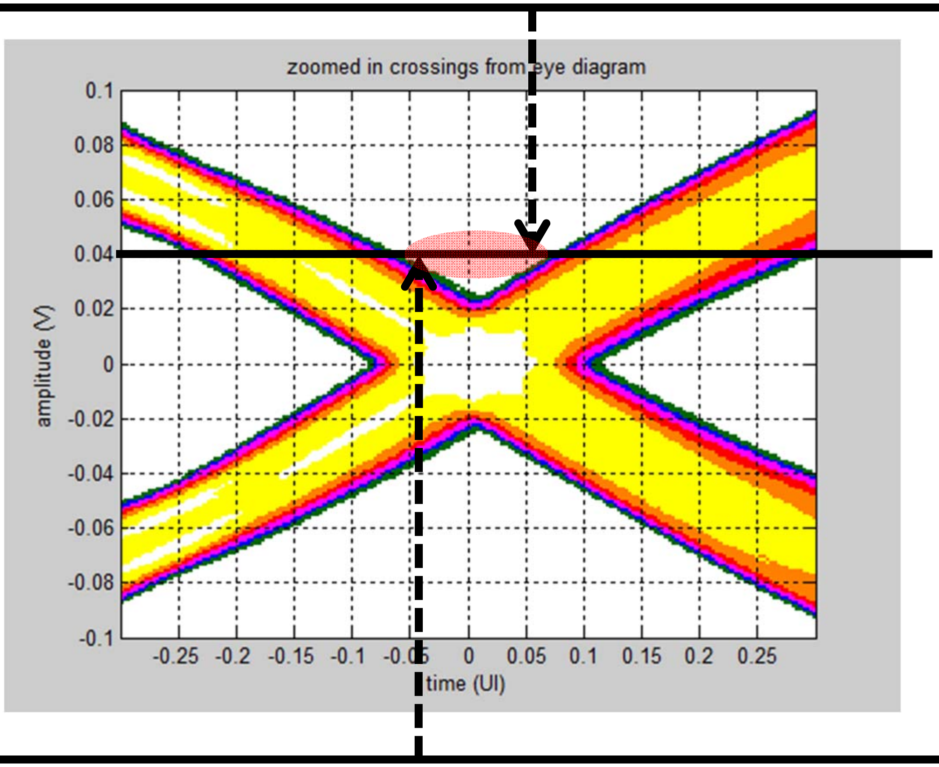
$$L(s) = \frac{K_{BB} I_{CP} K_{VCO}}{s}\left(R + \frac{1}{sC}\right)$$

Practical Issues with Bang Bang Phase Detectors

- Loop gain is proportional to the transition density (we derived for long term average)
  - Nyquist data (101010…) has a transition density of 1 (valid 8B10B data)
  - For 8B10B data we can have long term transition density as low as .3
    - For 8B10B coded sustainable long term gain can vary by a factor of 3.33:1
  - For un-coded data, transition density can be almost anything!
  - Later we will look at techniques to mitigate this problem
- Offsets in the phase slicer
  - We have assumed perfect slicing at 0
  - If offset then we are going to make some wrong decisions
  - Study this next

Offsets in Phase Slicer

Sampling offset is **late**, p_n always 0, thus **rising** edges always **early** **falling** edges always **late**. Averages to zero!

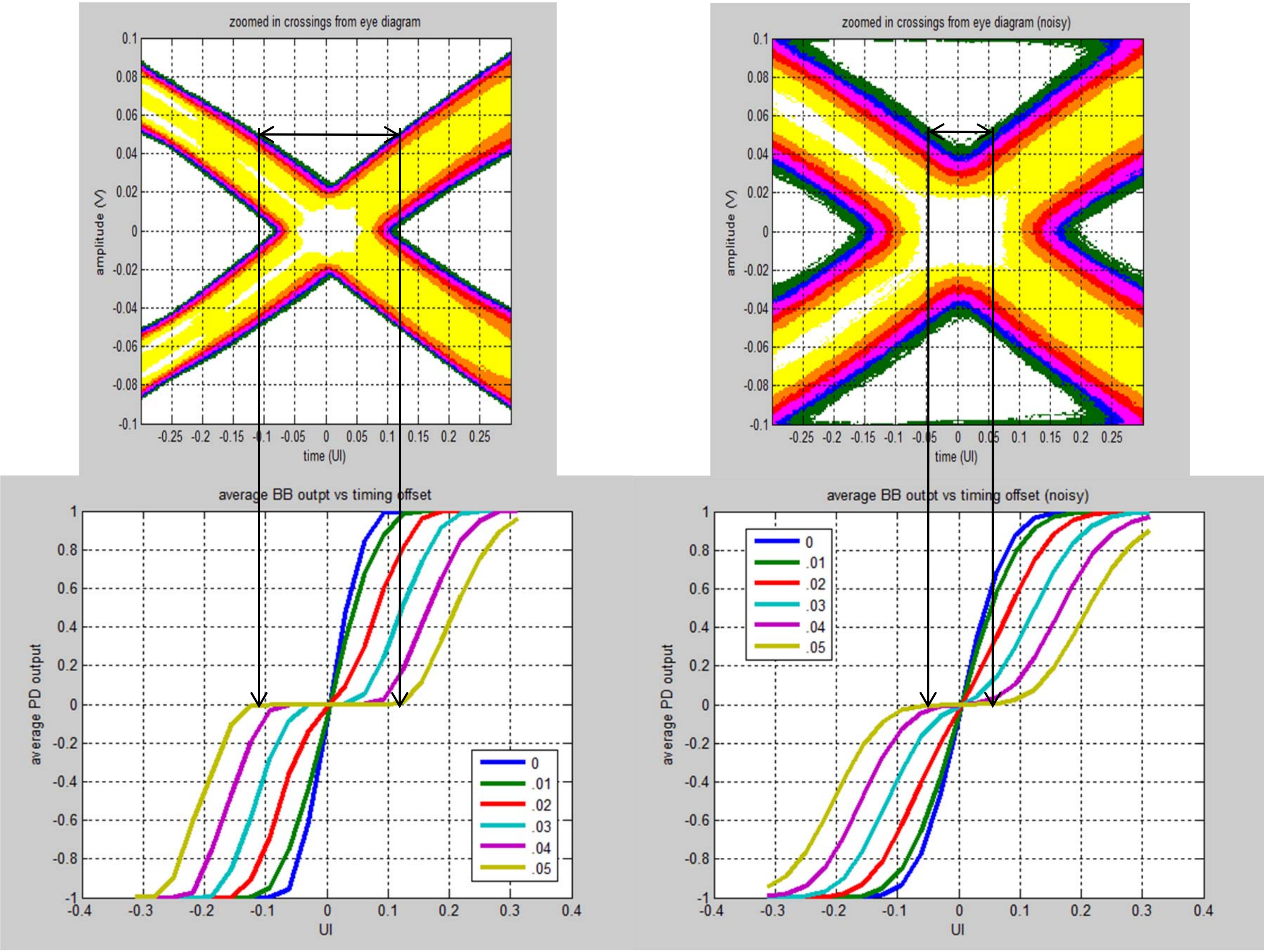


slicer offset

dead zone, **p_n always 0 whether early or late!!!**

Sampling offset is **early**, p always 0, but **falling** edges always **late** **rising** edges always **early**. Averages to 0!

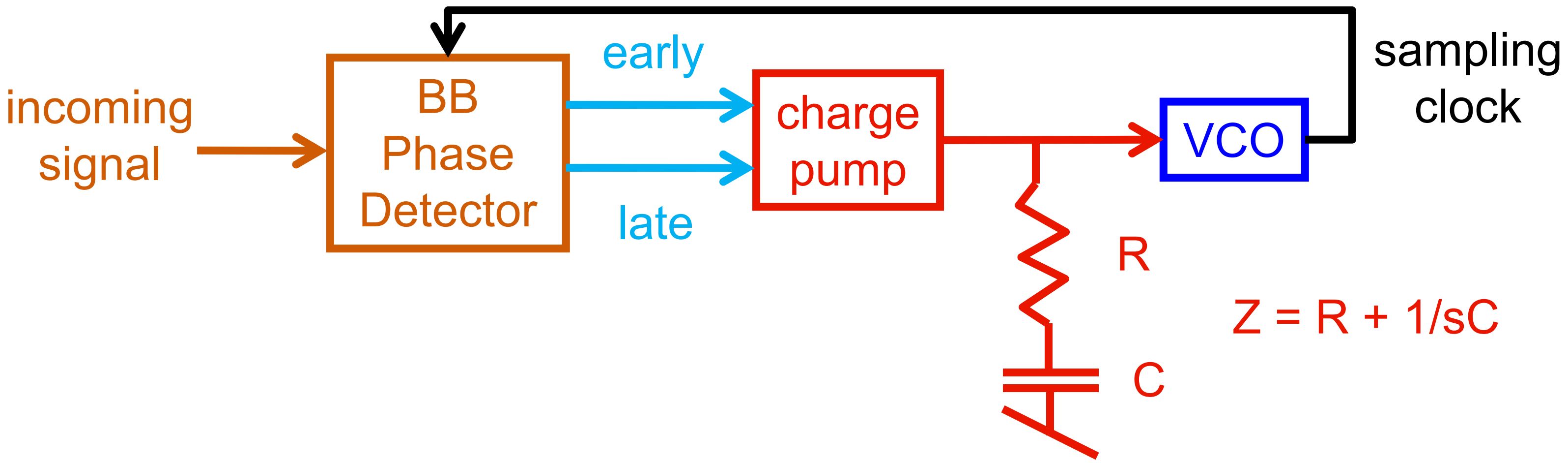
Compare Average BB Output

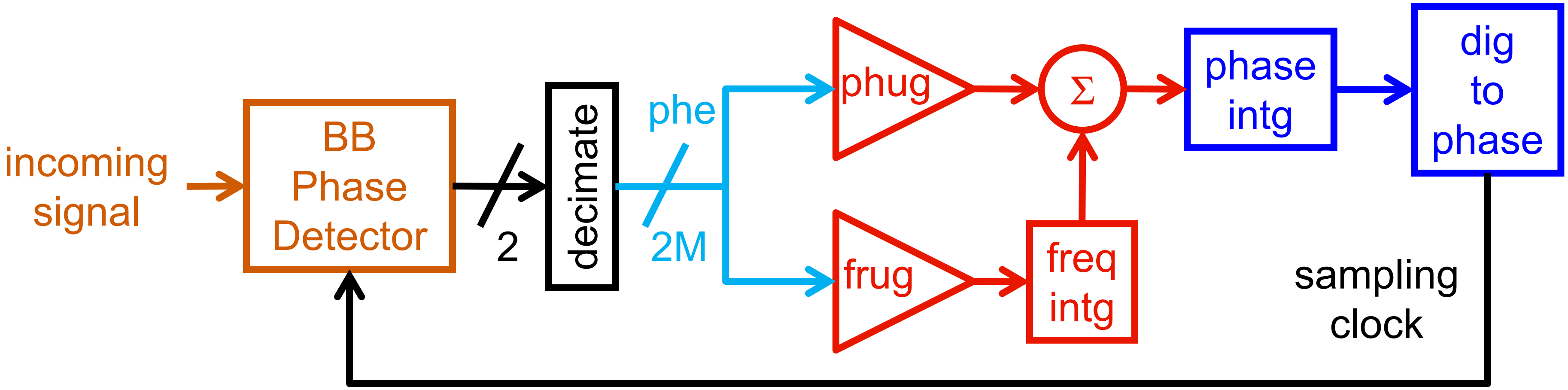


DPLL Based CDR

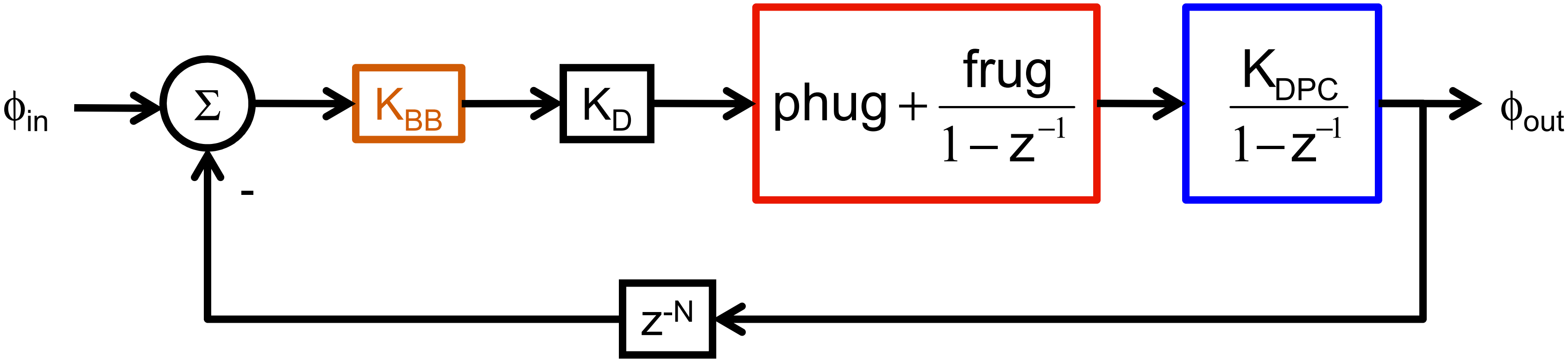
- Replace analog blocks in analog BB CDR with digital equivalents
  - BB phase detector stays the same
  - Digital charge pump and loop filter equivalent
  - Digital VCO equivalent
  - Analog block to convert digital to phase
- Derive small signal model and loop gain
- Justify design by comparing digital and analog small signal models

DPLL Structure





DPLL CDR Small Signal Model


ϕ_{in}
Σ
-
K_{BB}
K_D
$phug + \frac{frug}{1 - z^{-1}}$
$\frac{K_{DPC}}{1 - z^{-1}}$
ϕ_{out}
z^{-N}

model for delays

$$L(z^{-1}) = \frac{K_{BB} K_D K_{DPC}}{1 - z^{-1}} \left(phug + \frac{frug}{1 - z^{-1}} \right) z^{-N}$$

Loop Gain Comparison

$$L(s) = \frac{K_{BB}K_{VCO}}{s}\left(I_{CP} \cdot R + \frac{I_{CP}}{sC}\right)$$

$$L\left(z^{-1}\right) = \frac{K_{BB}K_{D}K_{DPC}}{1 - z^{-1}}\left(\text{phug} + \frac{\text{frug}}{1 - z^{-1}}\right)z^{-N}$$

phug ~ $I_{CP}R$

frug/(1-z^{-1}) ~ I_{CP}/(sC)

K_{PDC}/(1-z^{-1}) ~ K_{VCO}/s

K_{BB} ~ K_{BB}

K_D only in DPLL

DPLL Advantages

- DPLL based CDR derived from tried and true traditional analog BB CDR
- phug acts like an $I_{CP}R$ **with no PVT variation**
- frug/(1-z^{-1}) acts like a I_{CP}/sC **with no PVT variation**
- Phase integrator acts like VCO **with no PVT variation**
- DPLL Based CDR acts like an **ideal** analog CDR with **greatly reduced area** since capacitors and resistors replaced with digital gates
- Improved observability and testability

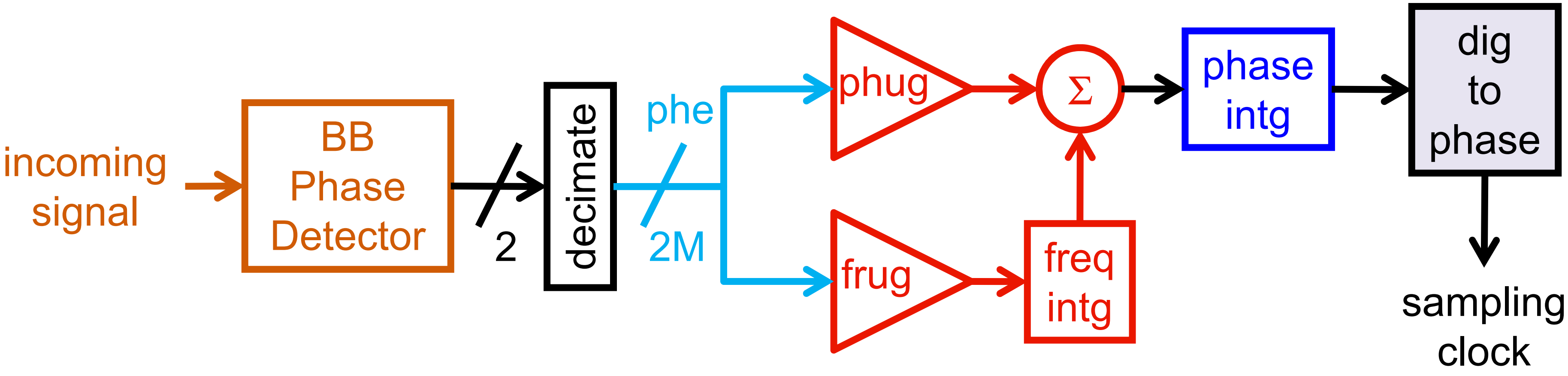
DPLL Design

- Linear model will provide us with insight in the design process
  - Right now parameters have no physical meaning
  - Parameters of the model need to be defined in terms of how they affect sampling phase
- To gain this understanding we need to walk backwards through the DPLL from the output sampling clock back to the BB PD output
  - Ultimately understand how BB PD output affects sampling phase as it passes through the system

DPLL Structure

$$L(z^{-1}) = \frac{K_{BB}K_{D}K_{DPC}}{1-z^{-1}}\left(\text{phug} + \frac{\text{frug}}{1-z^{-1}}\right)z^{-N}$$

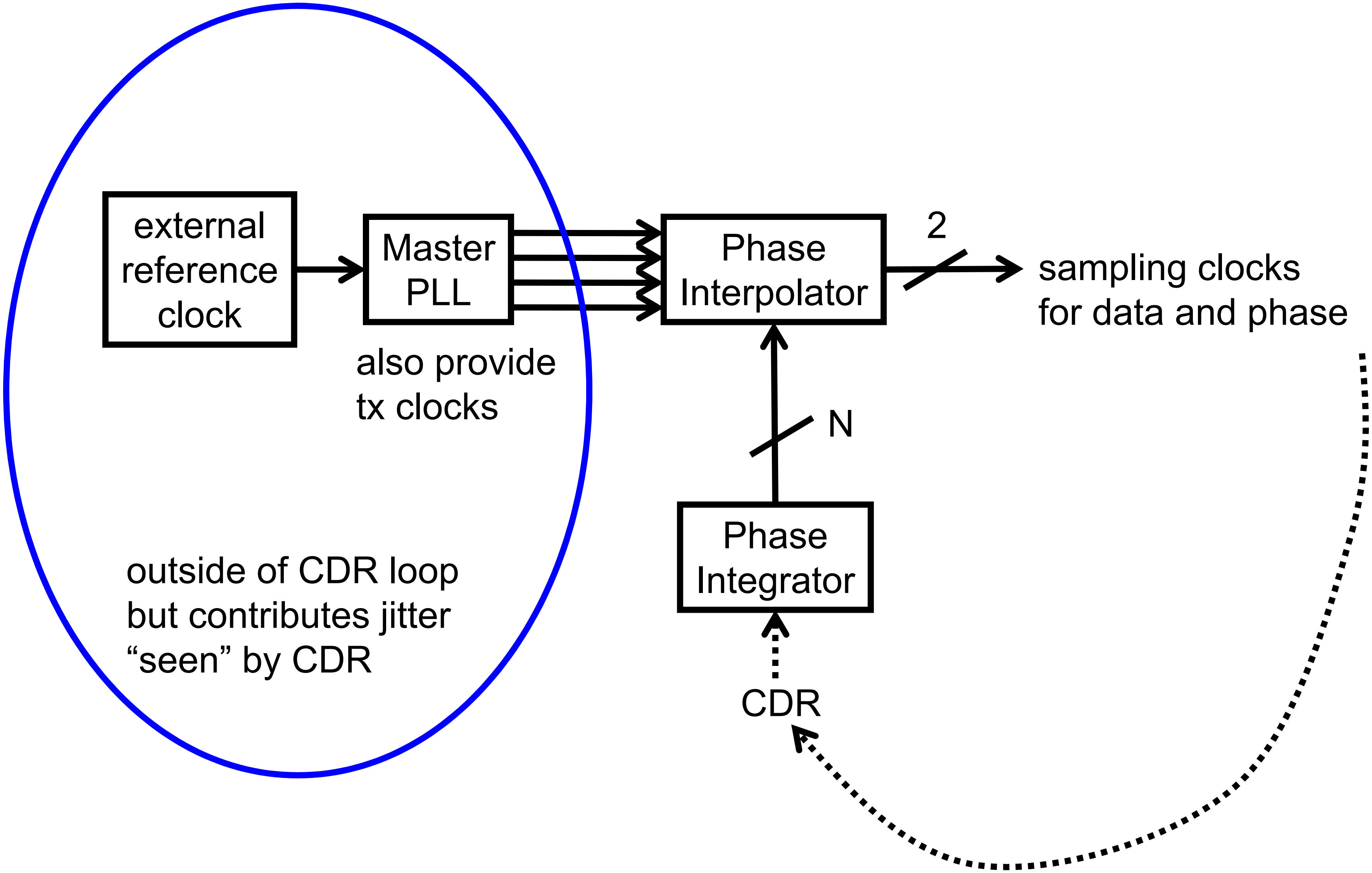
K_{DPC} = ?
phug = ?
frug = ?
K_D = ?



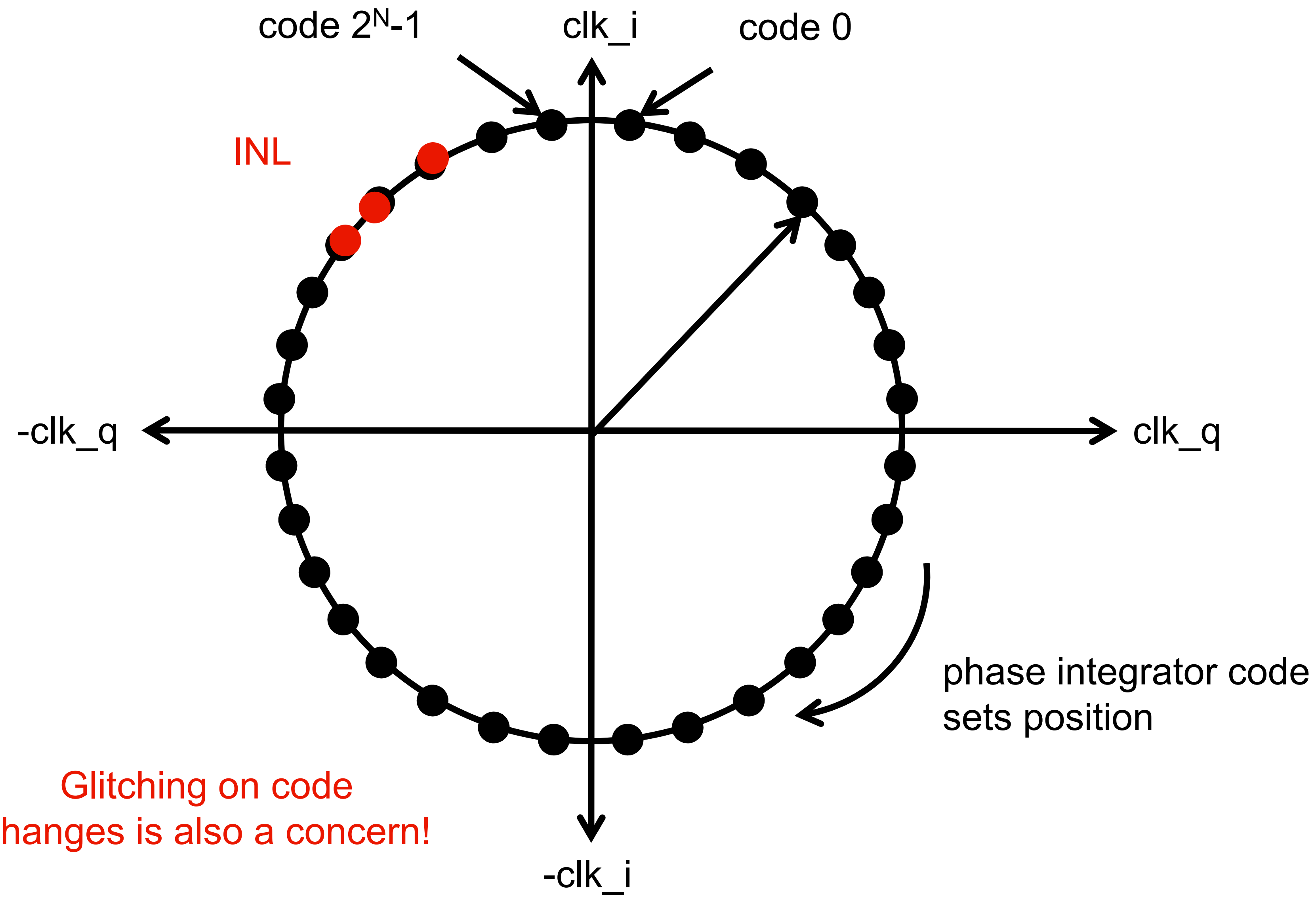
Digital Phase Converter

- Converts digital code from the phase integrator into an edge location with respect to the **local** reference clock
  - Phase Interpolator (PI), phase rotator, DLL, etc
- Design parameter of importance is resolution in bits. For N bits each UI is 2^N steps – discrete steps of **local** reference clock
  - There is quantization error vs analog CDR
  - Average PI will have 4 input (reference) phases and 32 output phase steps

Phase Interpolator Functional View

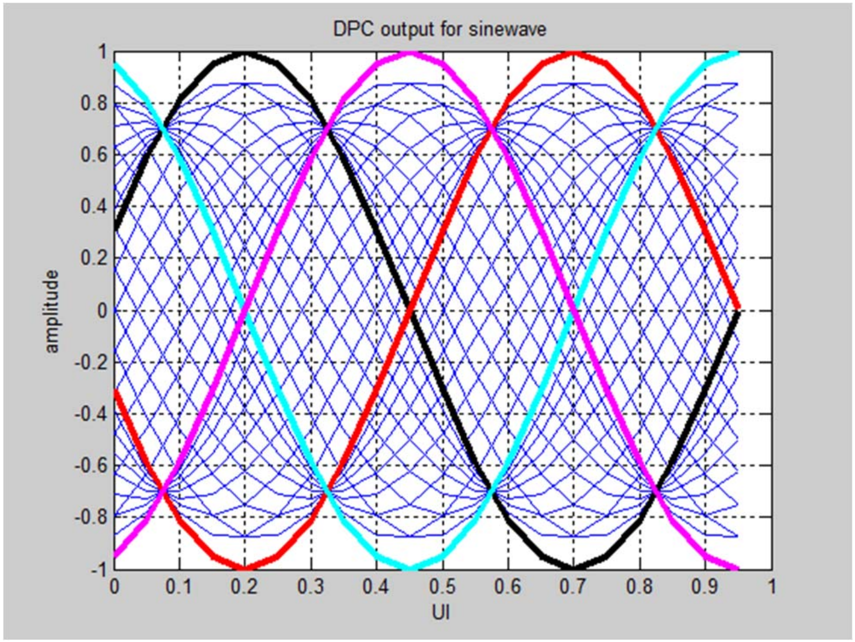


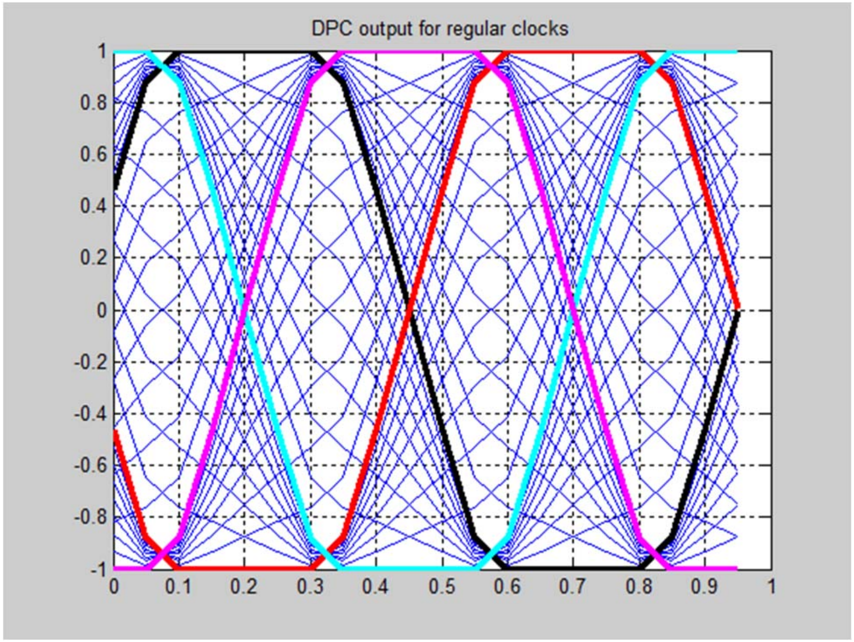
Phase Interpolator (Rotator)



Glitching on code changes is also a concern!

Impact of Reference Clock Edges on PI Output



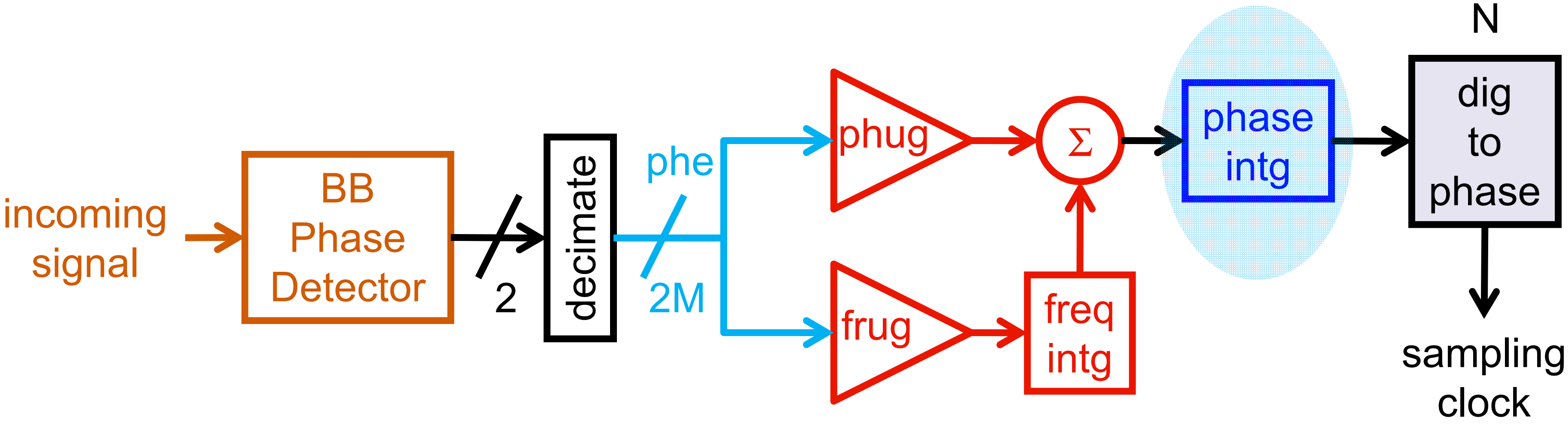


Challenge: Make reference clock edges as triangular shaped across PVT!!!

DPLL Structure

$$L\left(z^{-1}\right)=\frac{K_{BB}K_{D}K_{DPC}}{1-z^{-1}}\left(\text{phug}+\frac{\text{frug}}{1-z^{-1}}\right)z^{-N}$$

K_{DPC} = ?
phug = ?
frug = ?
K_D = ?



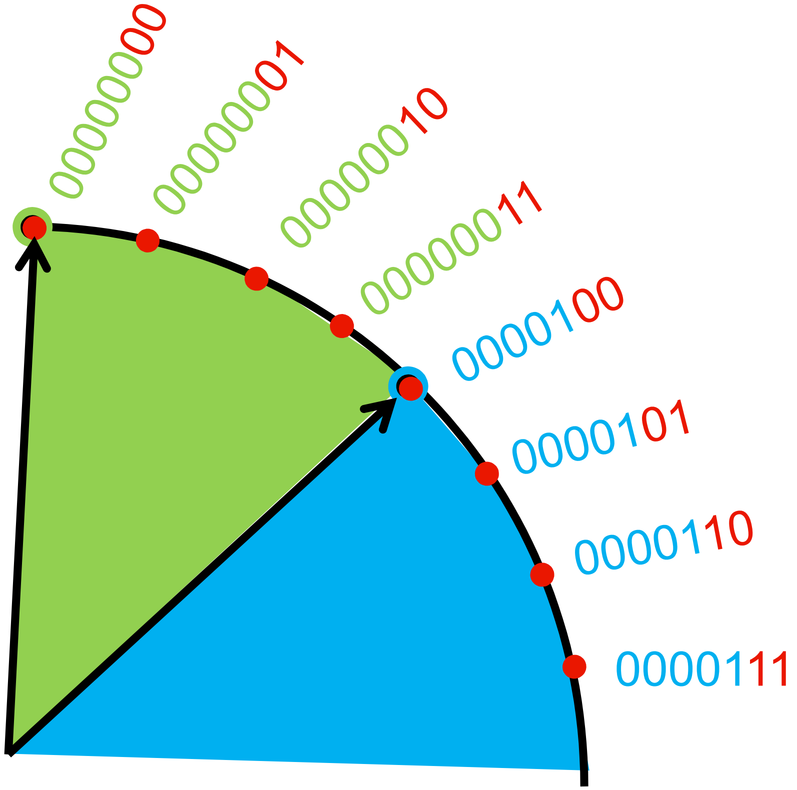
Phase Integrator

- Digital representation of the sampling phase via DPC
  - Phase integrator is N+D_p bits - each step in the phase integrator is effectively $1/2^{(N+D_p)}$ of a UI
  - **Sets the effective resolution for the system**, a value of 1 coming out of BBPD moves phase by $1/2^{(N+D_p)}$ of a UI
  - **$K_{DPC} = 1/2^{(N+D_p)}$** (Model of VCO gain)
- Phase integrator is unsigned (continuous phase rotation in PI)
  - max value is $2^{(N+D_p)}$ -1, for 8 bits range is 0 to 255
  - $(2^{(N+D_p)}$ -1)+ 1 = 0, for 8 bits 255+1=0
- If PI has 2^N steps then the top N bits of the phase integrator are attached to the PI
  - The D_p dither bits of phase integrator are sub-resolution
- Example: N=5, D_p=2
  - Phase integrator : 7’bxxxxxxx
  - Blue bits tied to PI, red bits effectively ignored

Phase Integrator Example 1

- N=5, D_p=2, PI and phase intg both 0, steps of 1 on each clock
  - Clock 0: phase intg = 7'b0000000
  - Clock 1: phase intg = 7'b0000001
  - Clock 2: phase intg = 7'b0000010
  - Clock 3: phase intg = 7'b0000011
  - Clock 4: phase intg = 7'b0000100
    - PI moves after 4 consecutive decisions
    - Effectively reduces max gain by a factor of 4

Graphical View

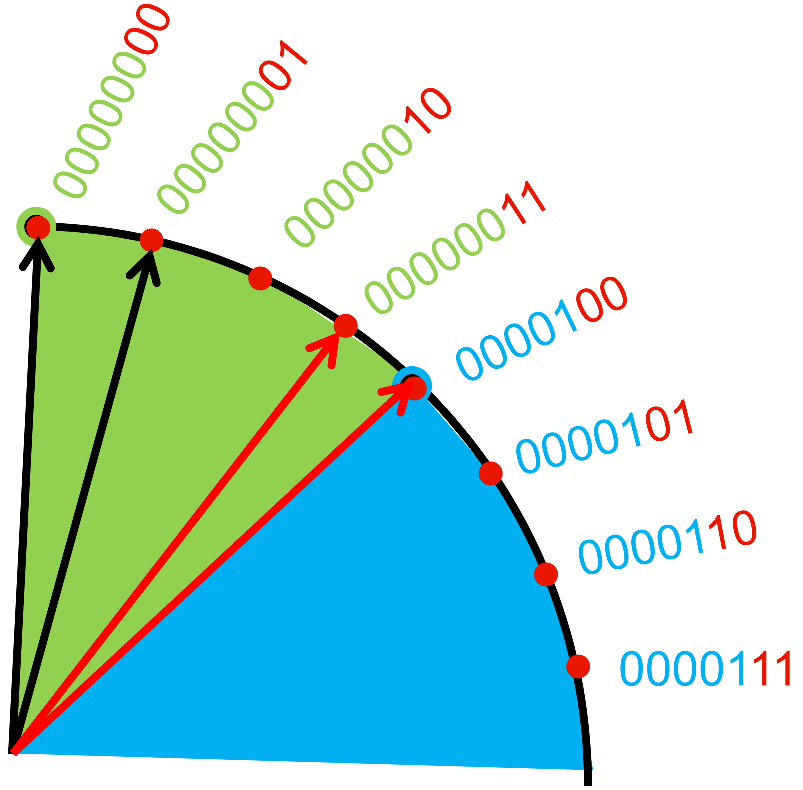


Red dots are the phase integrator values, colored regions are the PI regions with respect to phase integrator values.

Phase Integrator Example 2

- N=5, D_p=2, PI and phase intg both 0, dithering 1 and -1
  - – Clock 0: phase intg = 7'b0000000
  - – Clock 1: phase intg = 7'b0000001
  - – Clock 2: phase intg = 7'b0000000
  - – Clock 3: phase intg = 7'b0000001
  - – Clock 4: phase intg = 7'b0000000
    - PI did not move
  - – If on boundary PI will dither, but probability of that occurring reduced by $2^{-(D_p-1)}$

Graphical View

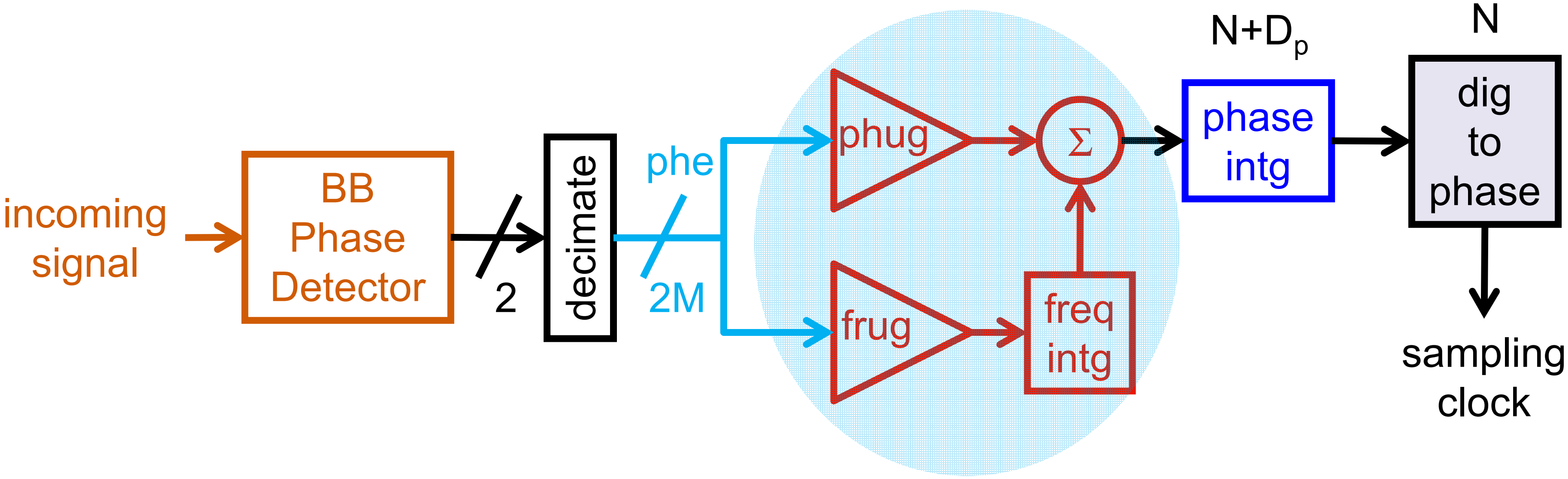


Dithering of output phase only occurs at boundaries, within boundaries dithering does not move the output clock

DPLL Structure

$$L\left(z^{-1}\right)=\frac{K_{BB}K_{D}K_{DPC}}{1-z^{-1}}\left(\text{phug}+\frac{\text{frug}}{1-z^{-1}}\right)z^{-N}$$

K_{DPC} = $1/2^{(N+D_p)}$
phug = ?
frug = ?
K_D = ?



Digital Loop Filter

- Filter the phase detector decisions (acts like CPLF)
- Proportional path through phug, integral path through frug
- phug and frug may be time-varying
  - At startup may want higher gain to increase pull-in range at cost of higher jitter
  - May turn on phug first to achieve a crude lock and then turn on frug
- Proportional path controls most of loop dynamics
  - Low latency in this path is extremely important
- Integral path is a slow path to compensate for long term frequency offset

Proportional Path (phug)

- Compensates for static phase offset with 0 steady state error
  - Same as proportional path in analog CDR
- Adds a +phug, -phug or 0 to phase integrator on each clock cycle
  - phug changes how much we move phase
  - Proportional path can move output phase by up to phug/$2^{(N+D_p)}$ of a UI per clock cycle
- phug is an integer value that affects how far phase moves in response to phase error
- phug usually in range of 0,1,2,4 (powers of 2 nice for multiplying digital values (shift)

Integral Path

- Compensate for a static frequency error with 0 steady state error
- A DC output from the frequency integrator is integrated by phase integrator and turns into a phase ramp
- If far end clock is 100PPM faster than local clock then it is 1+100/1e6=1.0001 faster
  - In 10000 of our local clocks the far end will send 10001 bits
  - To keep pace CDR needs to advance 1 extra UI every 10000UI
  - Freq intg should add 1/10000UI to phase integrator every UI
- Frequency integrator is signed!
  - 2’s complement arithmetic
  - Sign extended and added to phase integrator
- Frequency integrator saturates

Frequency Integrator Details

- Frequency integrator is $M+D_f$ bits with top M bits sent to phase integrator and D_f dither bits (fractional bits)
  - A 1 in the top M bits moves phase $1/2^{N+D_p}$ UI on each cycle
- Range of top M bits is -2^{M-1} to $2^{M-1}-1$
- Max frequency tracking:
  - If phase integrator is $N+D_p$ bits then frequency register can move sampling clock by a maximum of $(2^{M-1}-1)/2^{N+D_p}$ UI per cycle

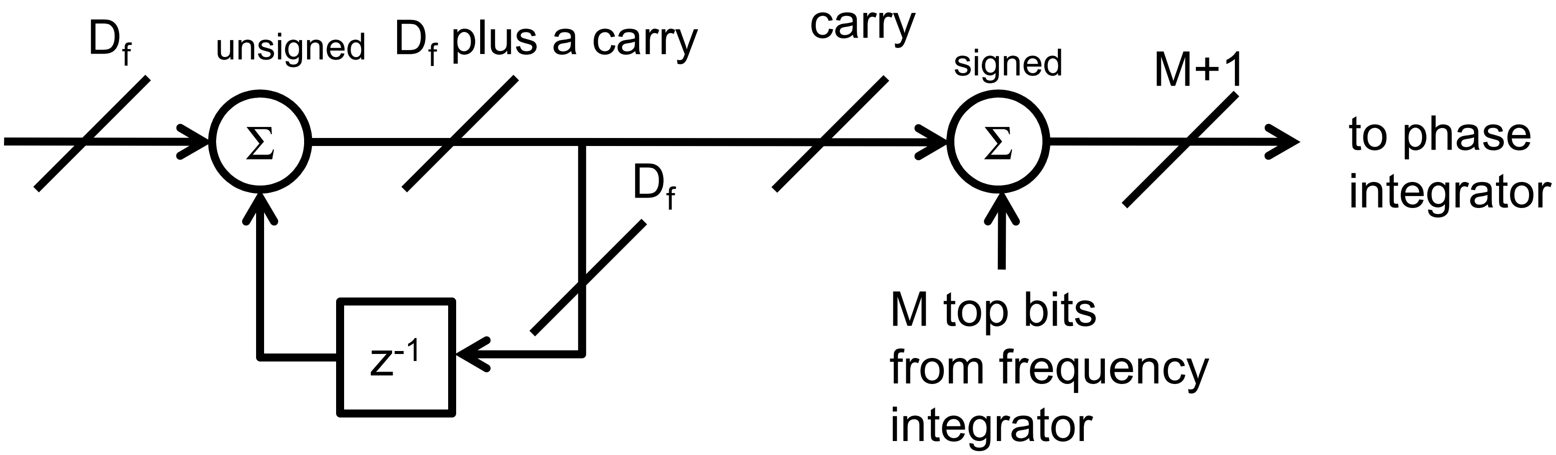
Frequency Integrator Example

- If phase integrator is 8 bits (1/256 UI) and frequency integrator output to phase integrator is a 1
  - Phase integrator 0, 1, 2, 3, …., 254, 255, 0, 1, …..
- Phase integrator moves 1/256UI per UI
  - This is moving phase by 1/256*1e6 = 3906PPM
  - Without sub-resolution we would only have a granularity of 3906PPM which is far too coarse
- To reduce granularity we need a way to have frequency integrator sub-resolution seen by phase integrator
  - Sub-resolution in phase integrator thrown on floor
  - Need to use sub resolution from freq intg

Frequency Integrator Sub-Resolution

- For a 100PPM offset we need to move the phase by 100UI every 1e6 UI (.0001UI per UI)
- Minimum sub-resolution step in an 8-bit phase intg is 1/256UI (.0039UI)
- If we added 1 to phase integrator every 39 UI we would advance phase an average of .0039UI/39UI = .0001UI/UI
- Dither bits can create a fractional input through **time averaging**

Sub-Resolution Time Averaging



On each cycle take bottom D_f bits of frequency register as input to an D_f bit integrator and the output is the carry bit

Acts like a D_f bit ΔΣ modulator

Output (carry) is either 0 or 1 – this is an unsigned output

Output is added to top M bits of frequency register in a signed add

Sub-Resolution Example: ¼

- Freq integrator=1/4: 7’b00000_01 (M=5, D_f=2)
- M=5’b00000, D_f =2’b01, ΔΣ state=3’b0_00
  - ΔΣ state = 3’b0_01, output to phase intg is 0+0 = 0
  - ΔΣ state = 3’b0_10, output to phase intg is 0+0 = 0
  - ΔΣ state = 3’b0_11, output to phase intg is 0+0 = 0
  - ΔΣ state = 3’b1_00, output to phase intg is 0+1 = 1
  - ΔΣ state = 3’b0_01, output to phase intg is 0+0 =0
  - …
  - Pattern repeats with a 1 being sent every 4th cycle = ¼ each cycle
  - Gain of integral path is frug/2^{Df}

Sub-Resolution Example: -¼

- Freq integrator=-1/4: 7’b11111_11 (M=5, D_f=2)
- M=5’b11111, D_f =2’b11, ΔΣ state=3’b0_00
  - ΔΣ state = 3’b0_11, output to phase intg is -1+0 = -1
  - ΔΣ state = 3’b1_10, output to phase intg is -1+1 = 0
  - ΔΣ state = 3’b1_01, output to phase intg is -1+1 = 0
  - ΔΣ state = 3’b1_00, output to phase intg is -1+1 = 0
  - ΔΣ state = 3’b0_11, output to phase intg is -1+0 = -1
  - …
  - Pattern repeats with a -1 being every 4th cycle = -¼
  - Unsigned fractional portion is +3/4, top bits are -1 together they add to -1/4

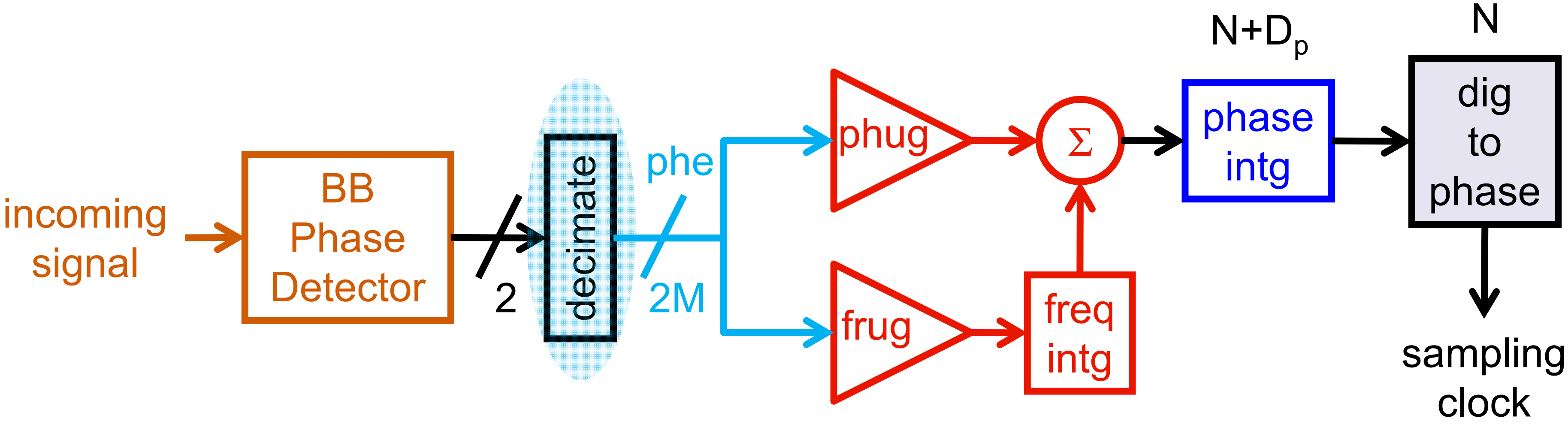
Frequency Integrator Summary

- Maximum PPM tracking is $(2^{M}-1)/2^{N+D_p}$ UI per cycle
  - For non-SSC usually about +/-1000PPM is adequate
  - For SSC usually about +/-8000PPM is adequate
- Effective resolution is $1/2^{N+D_p+D_f}$ per cycle
  - A 1 in the frequency register takes $2^{N+D_p+D_f}$ cycles to move 1 UI
  - In linear model frug value is replaced by frug/2^{D_f} (this will change once more when we talk about decimation)
- Maximum slew rate is given by
  - (frug*phe)/$2^{N+D_f+D_p}$
  - This is important for SSC tracking
- Dither bits can produce a fraction in the range of: $1/(2^{D_f})$ to $(2^{D_f} - 1)/2^{D_f}$

DPLL Structure

$$L\left(z^{-1}\right)=\frac{K_{BB}K_{D}K_{DPC}}{1-z^{-1}}\left(phug+\frac{frug}{1-z^{-1}}\right)z^{-N}$$

K_{DPC} = $1/2^{(N+D_p)}$
phug = {0,1,2,4}
frug = $\{0,1,2,4\}/2^{D_f}$
K_D = ?



Decimation

- Major power consumption in CDR from distributing high speed clocks
- Hard to make frequency and phase integrators run at baud rate
- Save power by slowing down and widening data path
  - Lookup table version of BB PD can be done on slower wider data if desired
  - Decimation for proportional and integral paths do not need to be the same
  - Decimation can be done in two ways: summing or voting
- Decimating by L means frequency register only added once every L UI, reduces integral gain by L thus integral path gain changes to frug/2^{D_f}/L in linear model

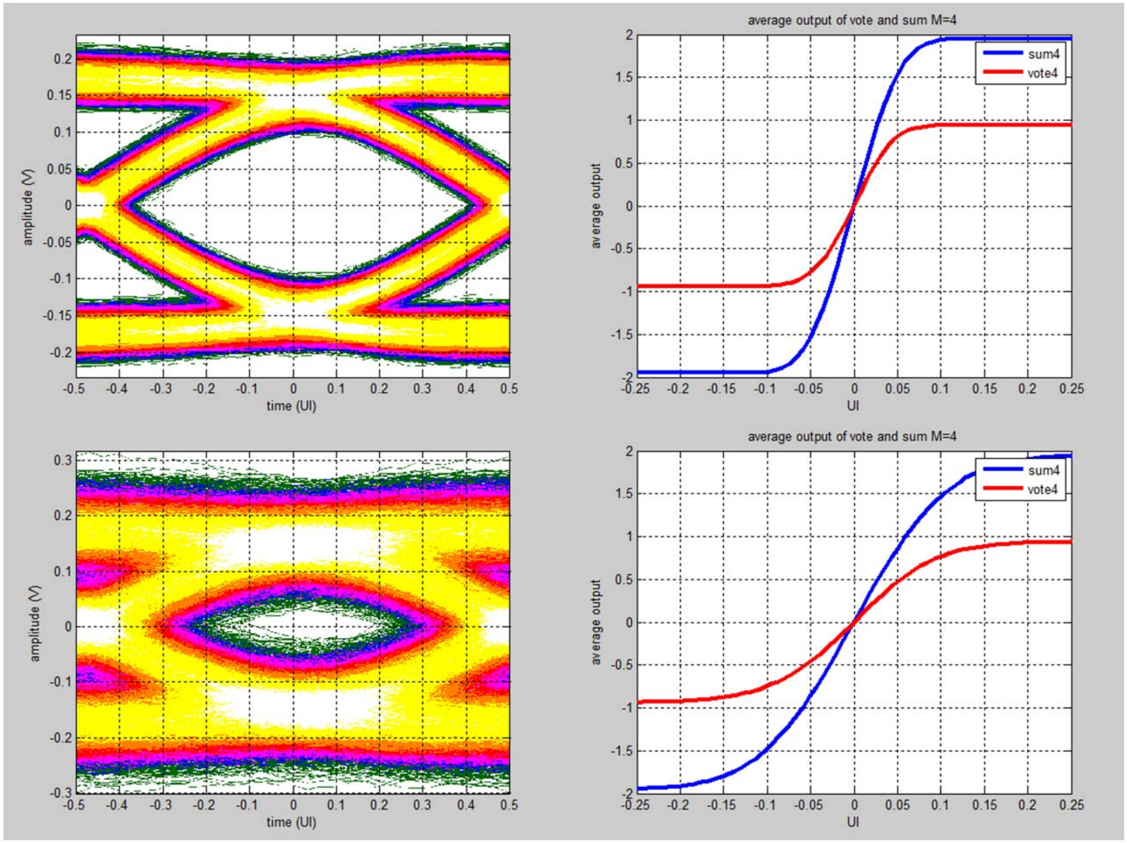
Decimation by Summing

- Add L phase error decisions (-1,0,1) together
  - No loss of information (acts like a pre-add)
  - Loop gain of proportional path is unchanged
  - In DSP this is called boxcar filter
  - Output is in range of –L:L once every L UI
  - 2L+1 possibilities log2(2L+1) bits wide at baud/L
  - Does not change loop sensitivity to transition density
  - **To reduce sensitivity to transition density we need to collapse multiple phase error decisions into a single decision**

Decimation by Voting

- Voting takes L samples and then decides either early or late based upon a majority decision (output is -1, 0 or 1)
  - Equivalent to taking the sign of summing decimation
  - Output is 2 bits wide at baud/L rate
- The –L:L range from summing is reduced to -1:1
  - Max gain reduced by factor of L
- Reduces dependence on transition density
  - If we are late and we have 1 transition per L UI or L transitions per L UI we get the same output
- We can mix the two techniques and vote over sub-blocks and sum together if desired
- Proportional path loop max loop gain reduced by 1/L
  - Small signal gain for linear model best derived by a simulation

Comparison of Decimation

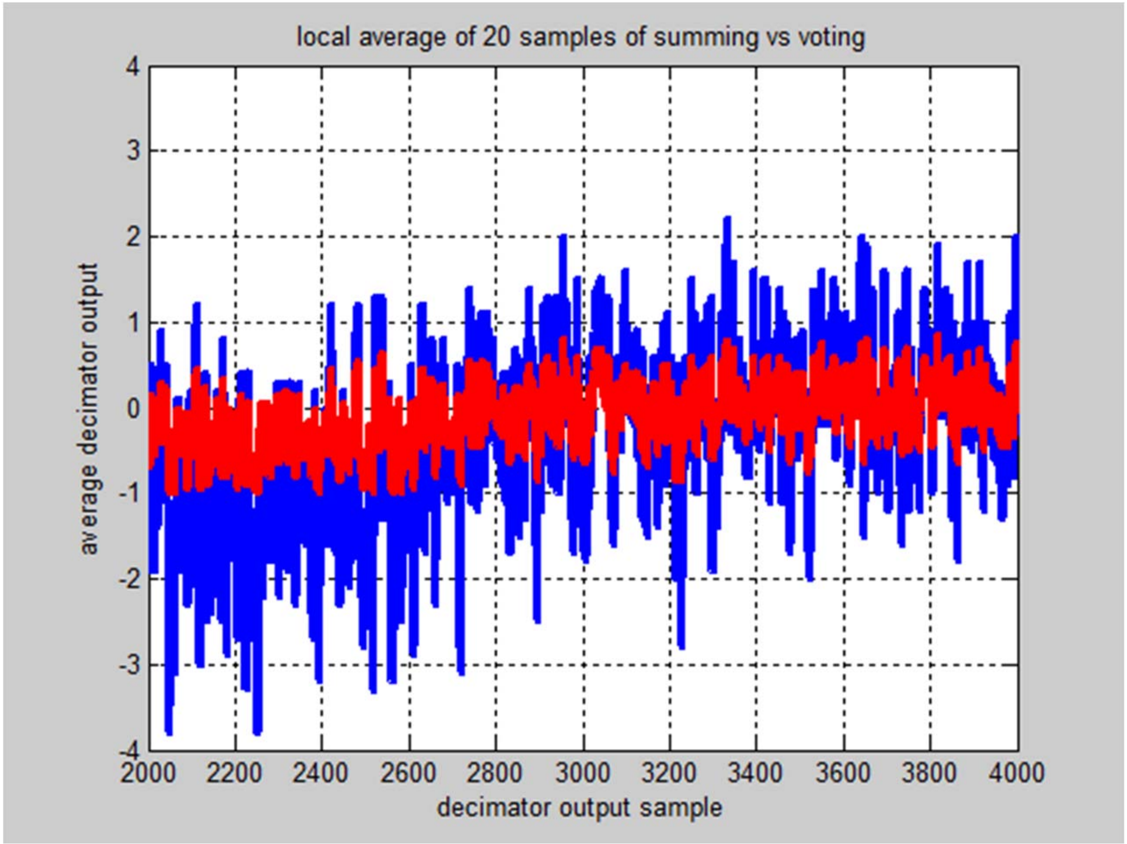


Both slope and maximum gain are reduced

Rule of thumb: Votingx4 has about ½ small signal gain of summing x4

Voting by different decimation factors will have different ratios

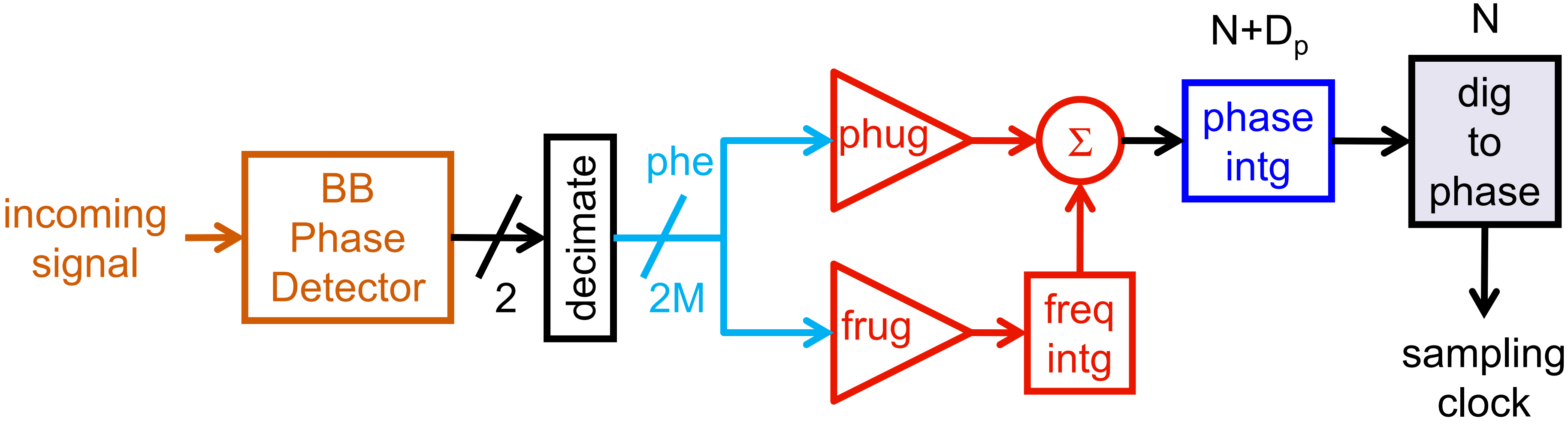
Comparison of Summing Versus Voting



DPLL Structure

$$L\left(z^{-1}\right)=\frac{K_{BB}K_{D}K_{DPC}}{1-z^{-1}}\left(phug+\frac{frug}{1-z^{-1}}\right)z^{-N}$$

K_{DPC} = $1/2^{(N+D_p)}$
phug = {0,1,2,4}
frug = {0,1,2,4}/(L*2^{Df})
K_D = .5 (vote x4), 1 (summing)



Latency (z^{-N})

- A major concern in DPLL based CDRs is the latency of the proportional path
- Latency limits the allowable loop gain and bandwidth
- Excessive latency leads to reduction in stability leads to peaking
- Need to account for all analog delay and digital pipe stages
  - If decimating by L then each digital pipe stage is L UI!
- Study impact later in design examples

Summary

- Developed a small signal model of DPLL
- Went through DPLL block by block to understand the detailed operation and impact of loop parameters
- Put it all together and see how we use this knowledge for design

CDR Example

- Baud rate is 5Gb/s
- ~ +/-1000PPM tolerance
- Frequency step of ~10PPM
- DPC is 5-bits (32 steps per UI)
  - Dither on a boundary is .03125UI - acceptable
- Phase integrator has 3 dither bits (8-bits total)
  - 1/256 = .0039UI/step is acceptable
- Decimate by voting over 4 decisions
  - CDR clock cycles are 4UI wide
- Assume latency around the loop is 20 UI (this includes delays for analog sections as well)

Max PPM Tolerance

- 1000 PPM is .001UI/UI
- CDR clock is 4UI (cycle is 4UI)
  - Frequency intg needs to supply .004UI/cycle
- Phase integrator step is 1/256 = .0039UI
  - Freq integrator needs to be able to put in +/-1 per cycle
- We need 1 signed upper bit to get to -1
  - M=1 (to represent either a 0 or a -1)
  - D_f bits will get us very close to +1: $(2^{D_f} - 1)/2^{D_f}$

Min PPM Step Size for D_f

- 10PPM resolution means moving sampling phase .00004UI/cycle
- 1 phase integrator step is .0039UI/cycle
- Compute the required number of cycles between dither contribution (1 in 2^{D_f})
  - .0039/.00004 = one step every 97.5 cycles
  - Round up to power of 2 yields 128 or 7 bits
  - D_f=7
  - Total frequency integrator width is $M+D_f = 8$

Final Frequency Register Results

- M = 1, D_f =7
- Maximum +PPM
  - (127/128)/256/4*1e6 = 972.5PPM
- Minimum -PPM
  - (-1)/256/4*1e6 = -976.6PPM
- Minimum PPM resolution
  - .0039/2^7/4UI * 1e6 = 7.6172 PPM < 10PPM

Pull-in Range

- Pull-in range is defined by how fast the proportional path can move in PPM
  - phug/256/4UI = phug/1024 per UI = 1/1024UI per UI
- This is equal to 976.6PPM **if we have a transition every 4 UI**
  - With 8B10B data at least 1 transition every 4UI thus we can achieve ~1000PPM pull-in
  - For uncoded data the pull-in range may be less!!!
- In general we can use larger phug at startup to increase pull-in range !

DPLL CDR Small Signal Model

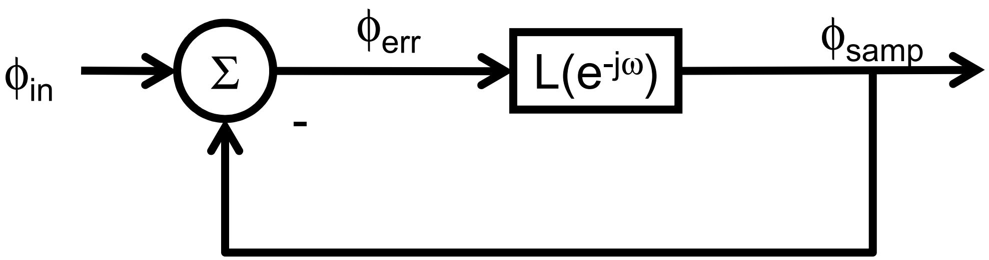
$$L\left(z^{-1}\right)=\frac{K_{BB}K_{D}\,2^{-\left(N+D_p\right)}}{1-z^{-1}}\left(\text{phug}+\frac{\text{frug}\cdot 2^{-D_f}\big/L}{1-z^{-1}}\right)z^{-N}$$

$$L\left(z^{-1}\right)=\frac{13.3\left(1\big/2\right)\cdot 2^{-8}}{1-z^{-1}}\left(\text{phug}+\frac{\text{frug}\cdot 2^{-7}\big/4}{1-z^{-1}}\right)z^{-N}$$

For analysis assume that input jitter is Gaussian with σ=.03UI

$$K_{BB}=\frac{1}{\sigma\sqrt{2\pi}}=\frac{1}{.03\sqrt{2\pi}}=13.3$$

Using the Linear Model

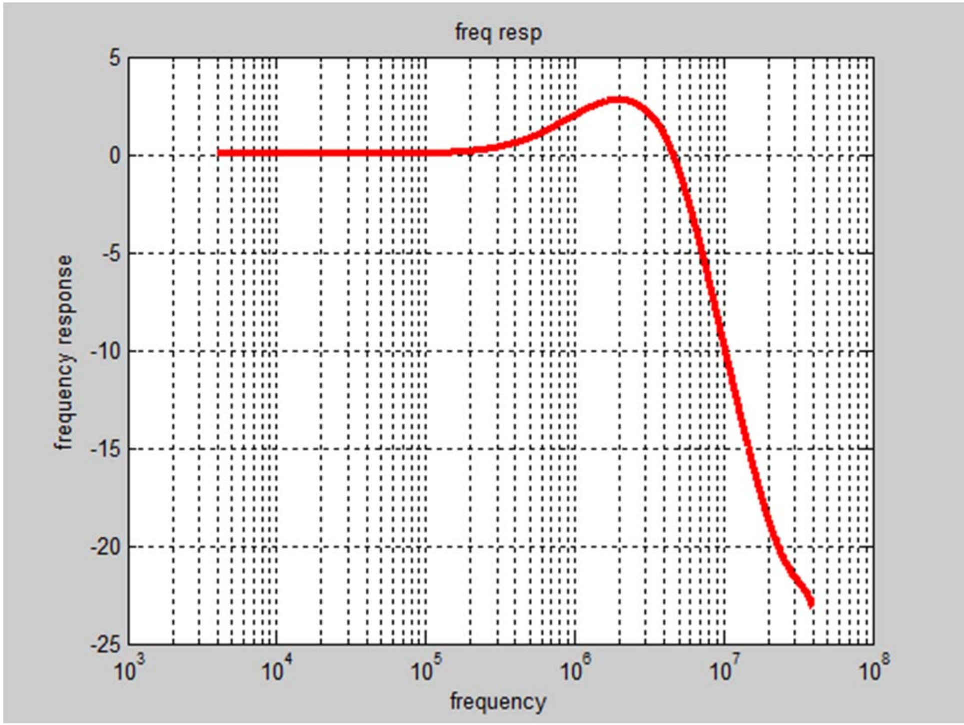


$$H\left(e^{-j\omega}\right) = \frac{\phi_{samp}}{\phi_{in}} = \frac{L\left(e^{-j\omega}\right)}{1 + L\left(e^{-j\omega}\right)}$$

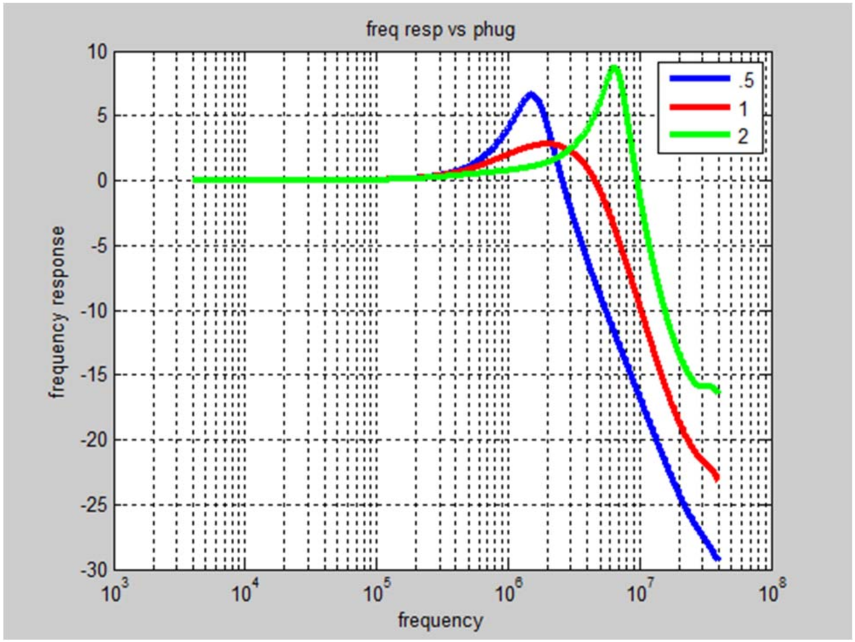
$$JT\left(e^{-j\omega}\right) = \left(\frac{\phi_{in}}{\phi_{err}}\right) \bullet \left(1 - \frac{12\sigma}{T_{UI}}\right) = \left[1 + L\left(e^{-j\omega}\right)\right] \bullet \left(1 - \frac{12\sigma}{T_{UI}}\right)$$

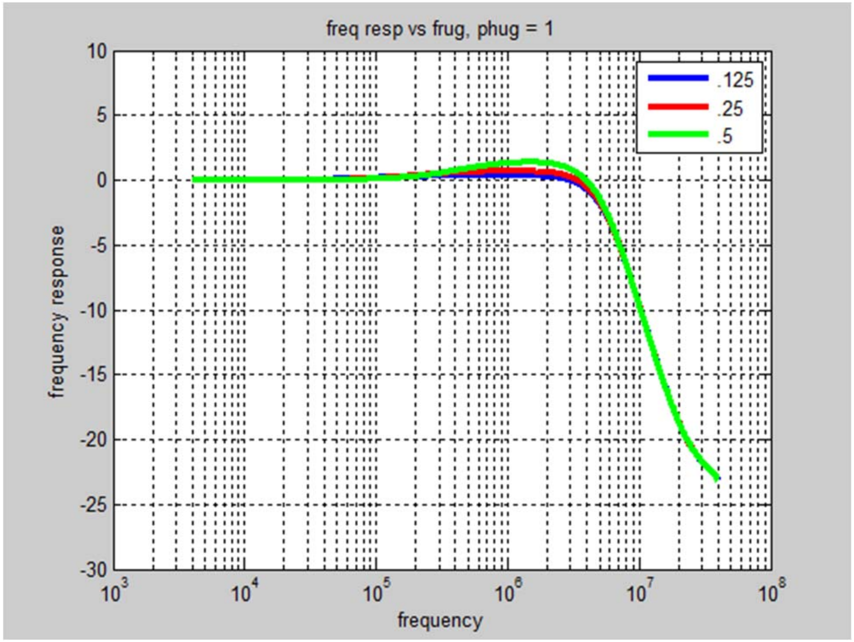
Transfer Function

$$H\left(e^{-j\omega}\right) = \frac{\phi_{\text{samp}}}{\phi_{\text{in}}} = \frac{L\left(e^{-j\omega}\right)}{1 + L\left(e^{-j\omega}\right)}$$



Study Peaking as a Function of phug and frug





lower frug looks like right choice, really lower integral path gain

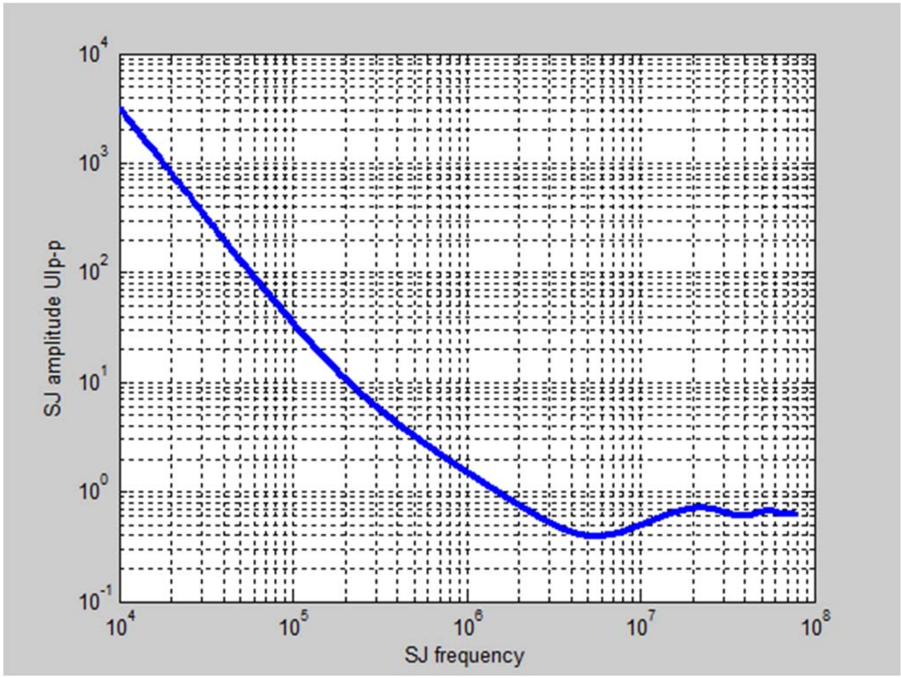
Final decision, lower integral path gain by .25

Integral Path Gain Reduction

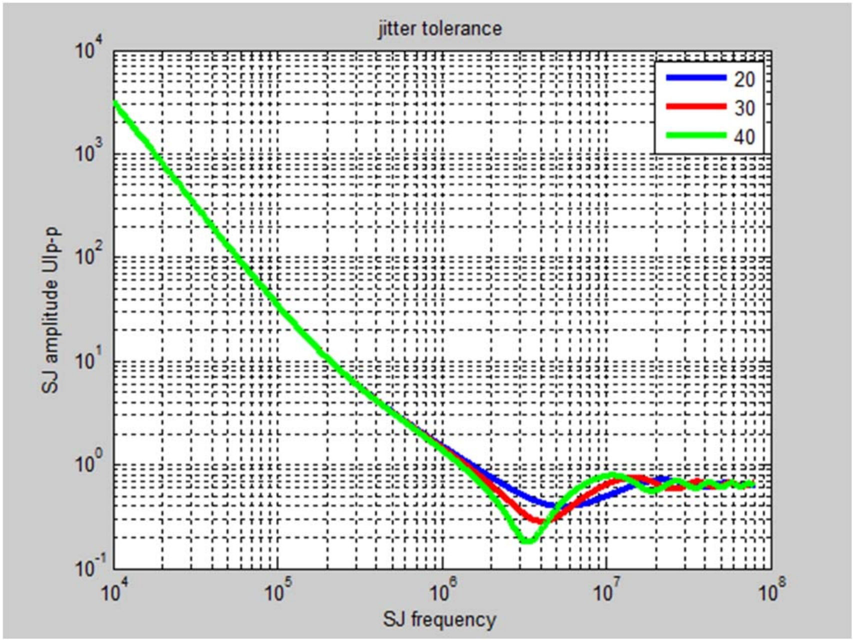
- Integral path gain = frug/2^{Df}/L
  - Either increase Df or increase decimation (L) for integral path
- Decimate frequency path by 16 instead of 4 (vote x16) decreases integral path by a factor of 4
  - Frequency integrator output only changes every 16UI, rate at which we can slew integrator reduced
  - Frequency integrator output still added to phase integrator every 4UI - PPM range not affected
- Frequency integrator was 8 bits at 1.25GHz now it is 8 bits at 312.5MHz
  - Fixed peaking and easier to implement!
- Minimum PPM resolution
  - .0039/2^7/16 * 1e6 = 1.9043 PPM

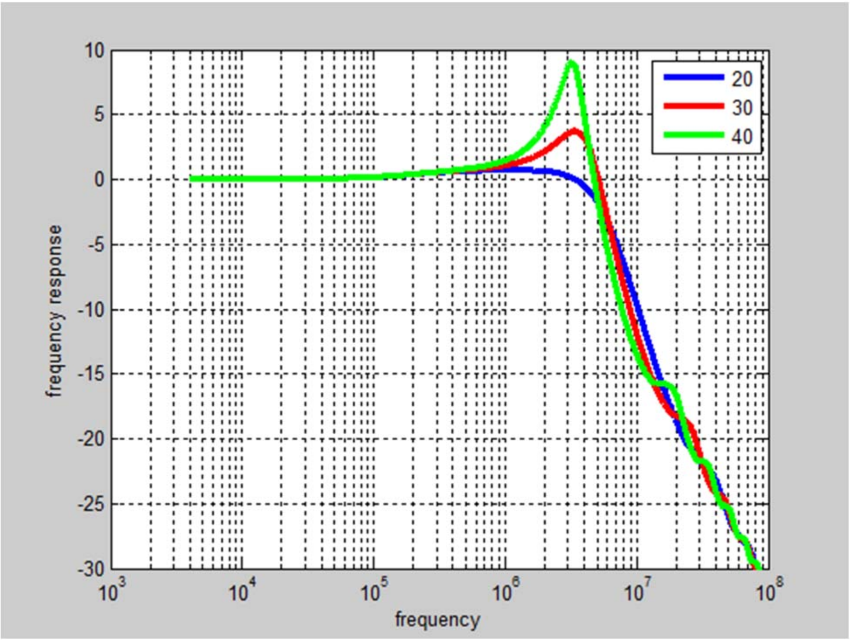
Jitter Tolerance with Reduced Integral Path Gain

$$\mathrm{JT}\left(e^{-j\omega}\right)=\left(\frac{\phi_{\mathrm{in}}}{\phi_{\mathrm{err}}}\right)\bullet\left(1-\frac{12\sigma}{T_{\mathrm{UI}}}\right)=\left[1+L\left(e^{-j\omega}\right)\right]\bullet\left(1-\frac{12\sigma}{T_{\mathrm{UI}}}\right)$$



Impact of Latency



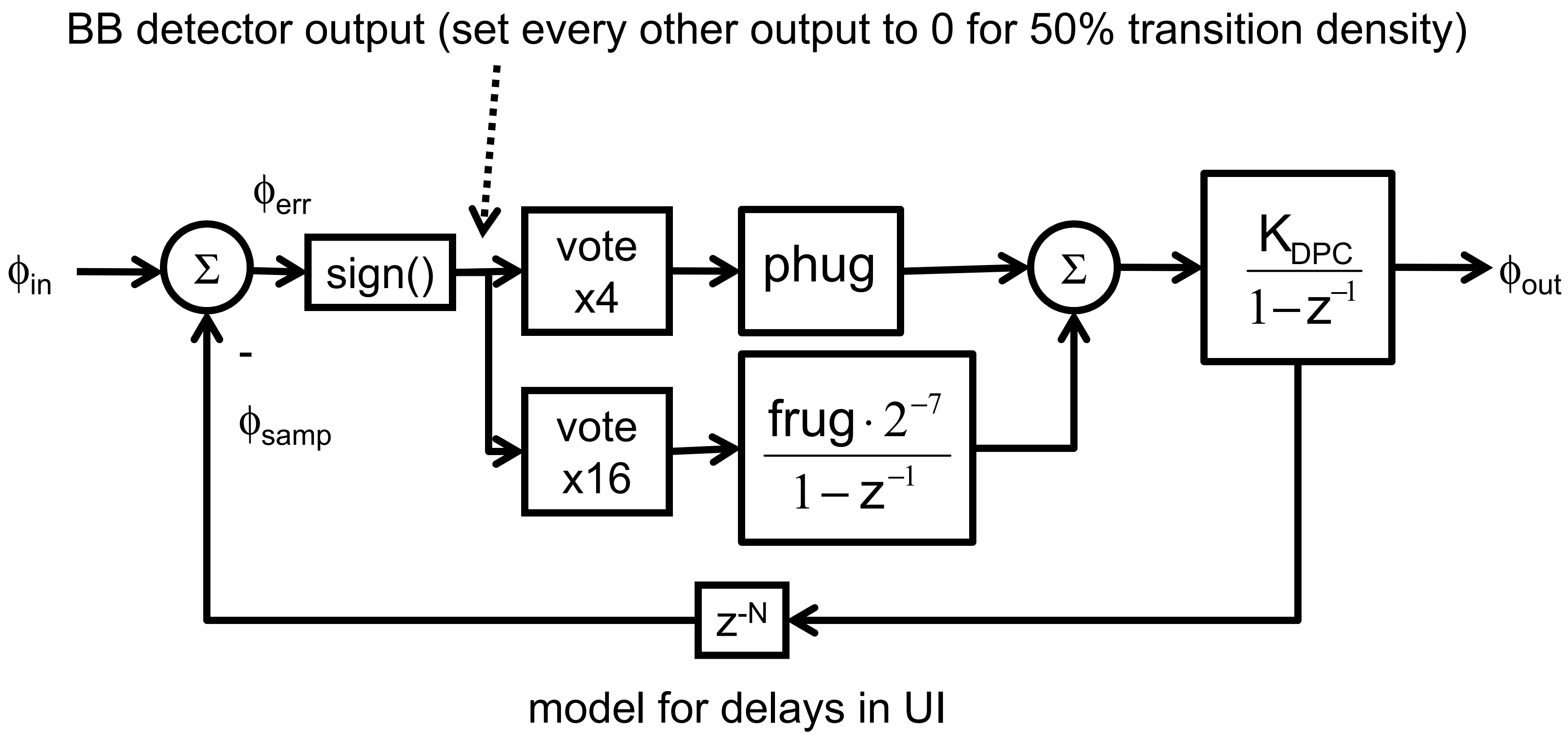


Latency needs to be controlled, a value of 20 is OK this design. If we can not meet latency then we need to lower gains = lower CDR bandwidth

Next Step in Design Process

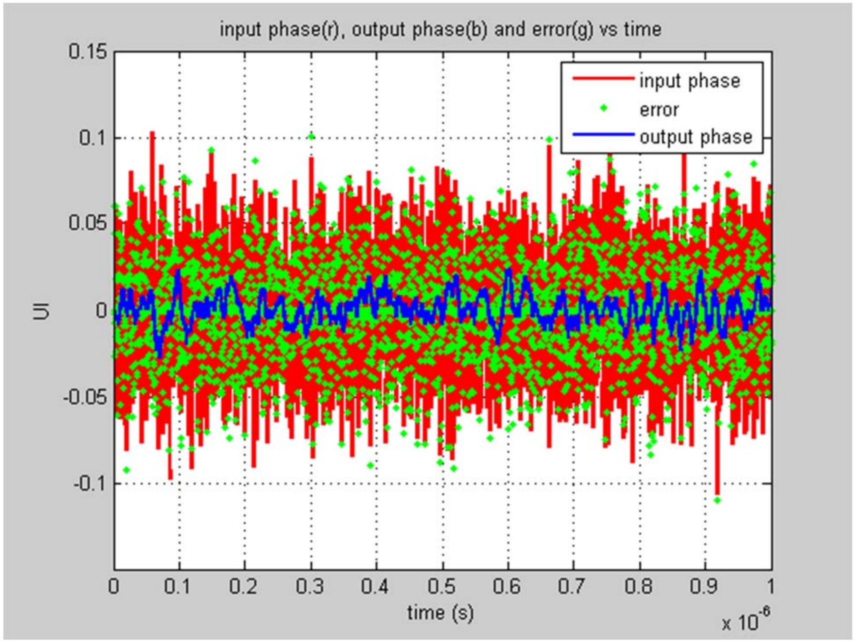
- Linear design provides much insight and a good starting point
- Need to check design based on linear model with a simple time step simulation
    - Start simple (no circuit distortions)
    - As design is refined add simulation complexity
    - Ultimately a very detailed system simulation will need to be run to verify final design

Simple Time Step Simulator

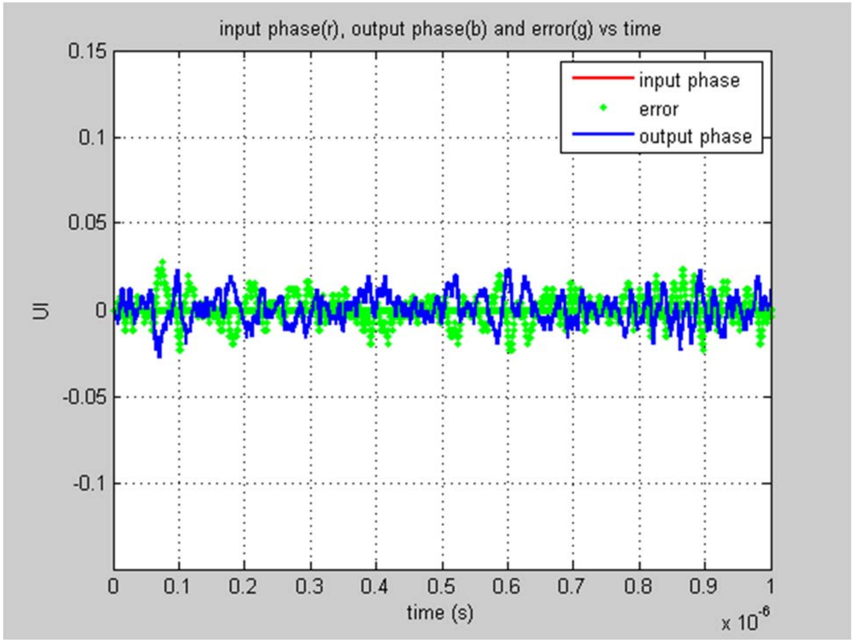

BB detector output (set every other output to 0 for 50% transition density)
ϕ_{in}
Σ
ϕ_{err}
sign()
-
ϕ_{samp}
vote x4
vote x16
phug
$\dfrac{\text{frug}\cdot 2^{-7}}{1-z^{-1}}$
Σ
$\dfrac{K_{DPC}}{1-z^{-1}}$
ϕ_{out}
z^{-N}
model for delays in UI

Time Step Results

actual simulation results



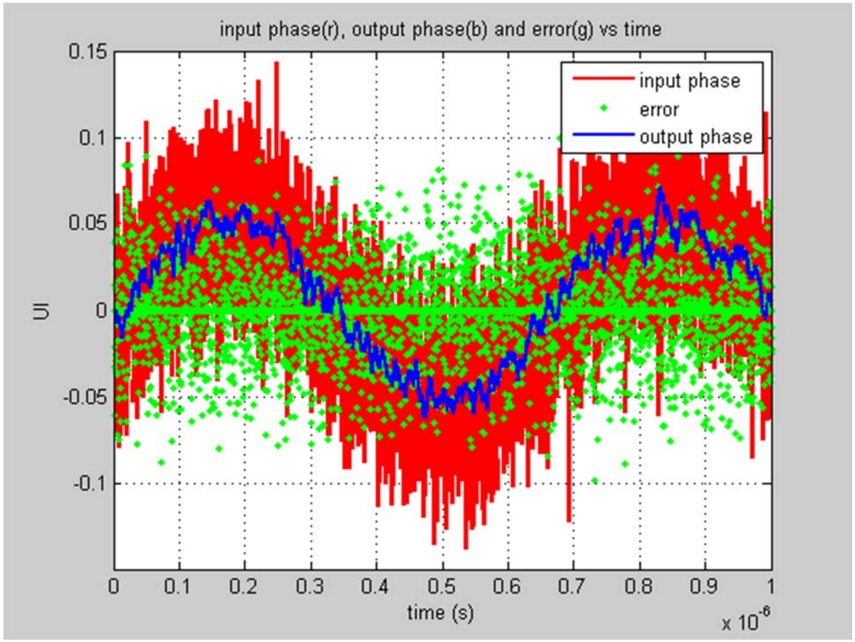
noise in simulation but removed from plot to focus on tracking



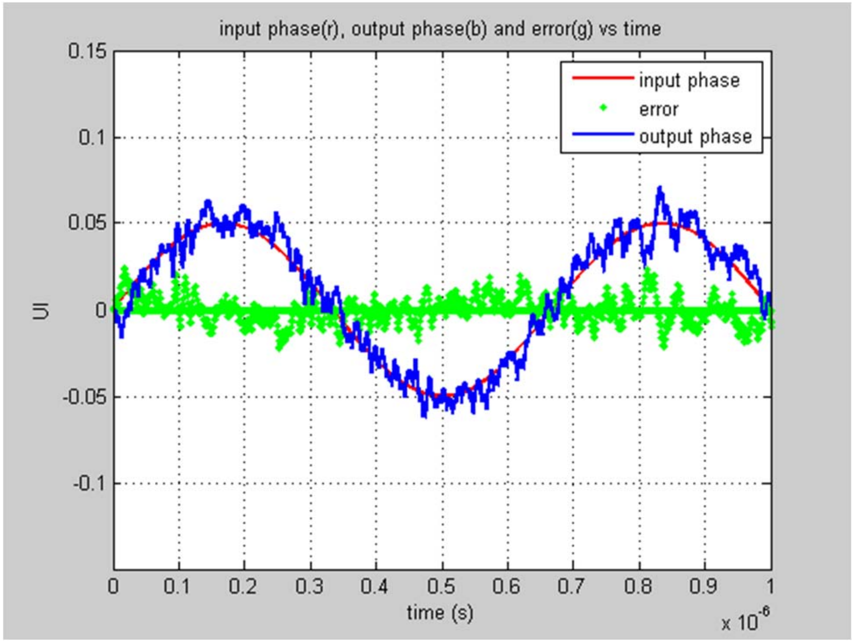
input phase noise is Gaussian, σ=.03, white out to 2.5GHz

Time Step Results, Small Signal SJ (.1UIp-p @ 1.5MHz)

actual simulation results



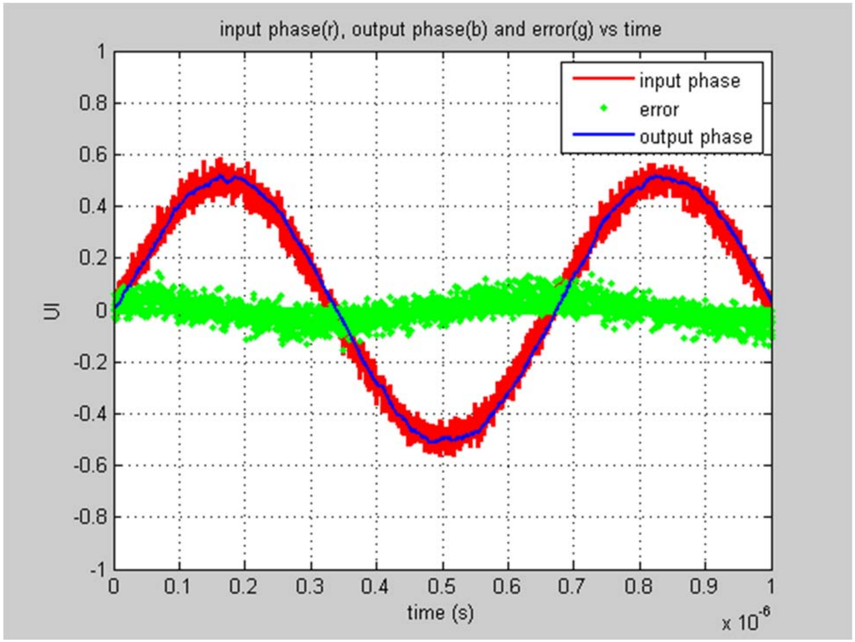
noise in simulation but removed from plot to focus on SJ tracking



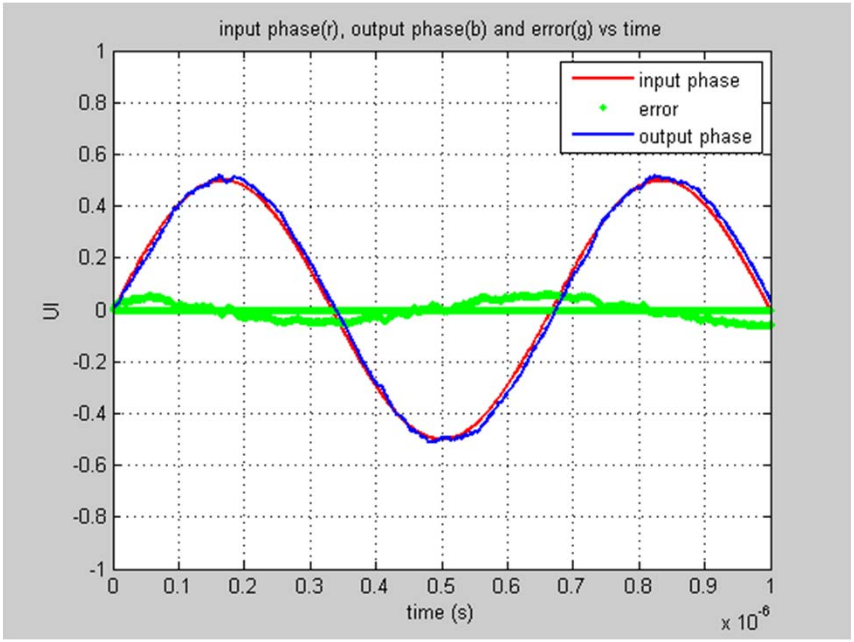
SJ of .1UIp-p at 1.5MHz plus input phase noise of Gaussian, σ=.03, white out to 2.5GHz

Time Step Results, Larger SJ, (1UIp-p @ 1.5MHz)

actual simulation results



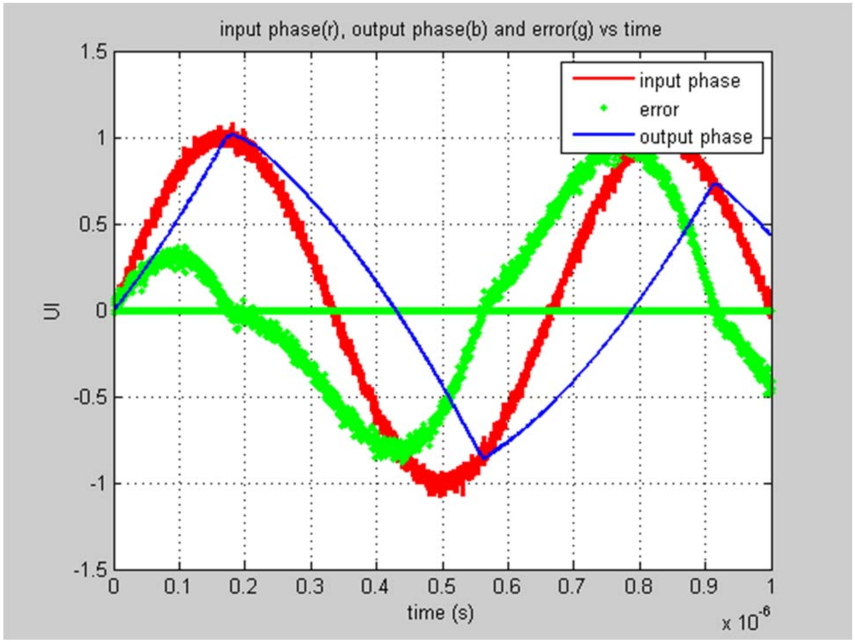
noise in simulation but removed from plot to focus on SJ tracking



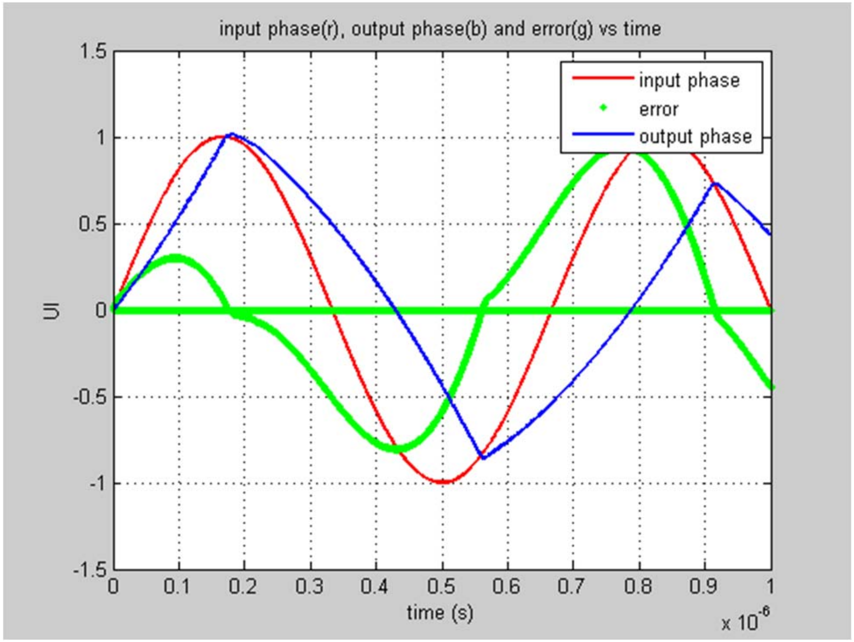
SJ of 1UIp-p at 1.5MHz plus input phase noise of Gaussian, σ=.03, white out to 2.5GHz

Time Step Results, Large Signal SJ, (2UIp-p @ 1.5MHz)

actual simulation results

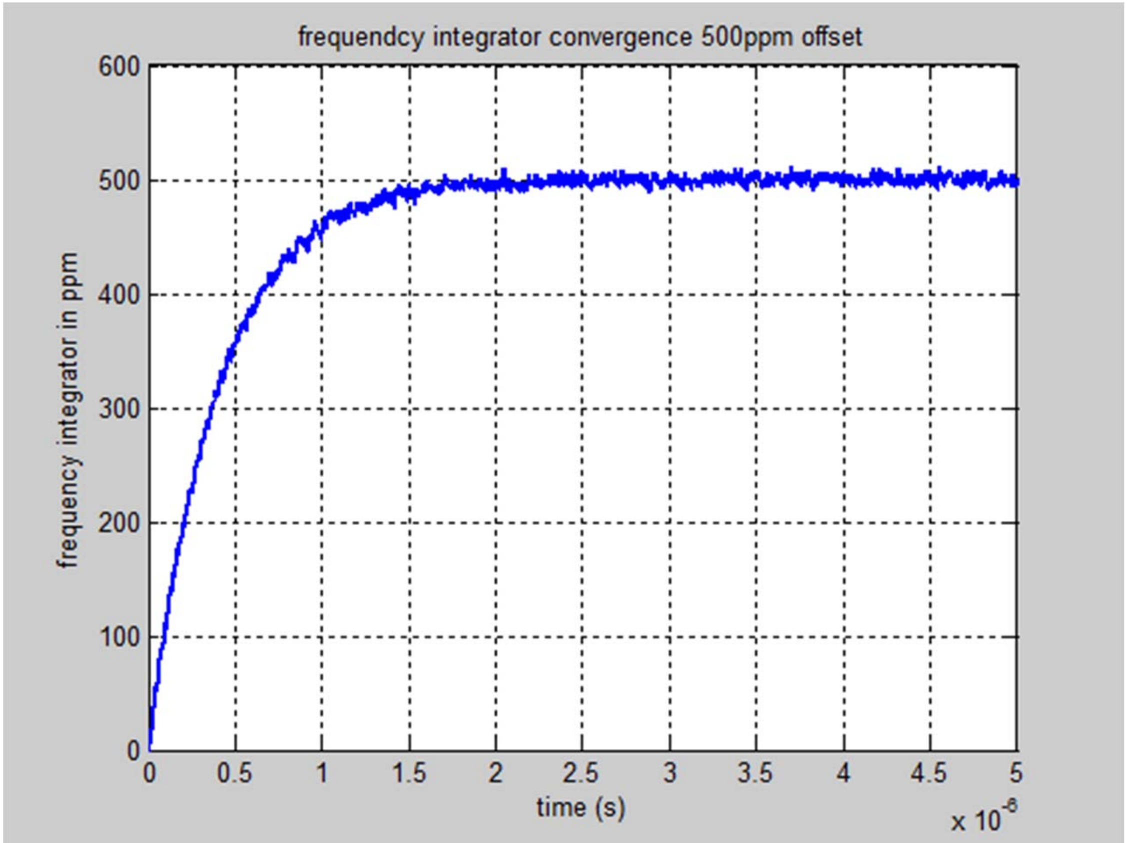


noise in simulation but removed from plot to focus on SJ tracking



SJ of 2UIp-p at 1.5MHz plus input phase noise of Gaussian, σ=.03, white out to 2.5GHz

Time Step Results, 500 PPM Offset



Design performance is reasonable, hold off on JT for next design

Spread Spectrum Clocking (SSC)

- Asynchronous Spread Spectrum Clocking (SSC)
  - Clock frequency is slewed to reduce EMI
  - Found in USB3, SATA and others
- Greatly increases the PPM tracking range requirement for the CDR
- New criterion: PPM slew rate
  - 2^{nd} order DPLL is underdetermined and tracking will be done with a steady state error

SSC Example

- Add SSC requirements to previous design
- Baud rate is 5Gb/s
- ~ +/-7000PPM tolerance
- Frequency integrator slew rate > 2000PPM/µs
- DPC is 5-bits (32 steps per UI)
  - Dither on a boundary is .03125UI - acceptable
- Phase integrator has 3 dither bits (8-bits total)
  - 1/256 = .0039
- Decimate phase path by voting over 4 decisions
  - CDR clock cycles are 4UI wide
- Decimate frequency path by voting over 16 decisions
- Assume latency around the loop is 20 UI (this includes delays for analog sections as well)

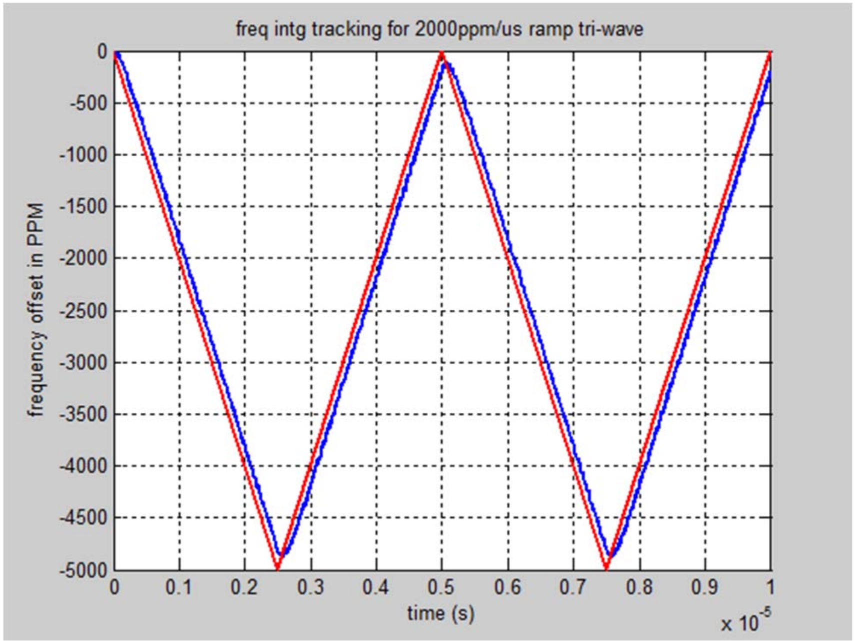
Max PPM Tolerance

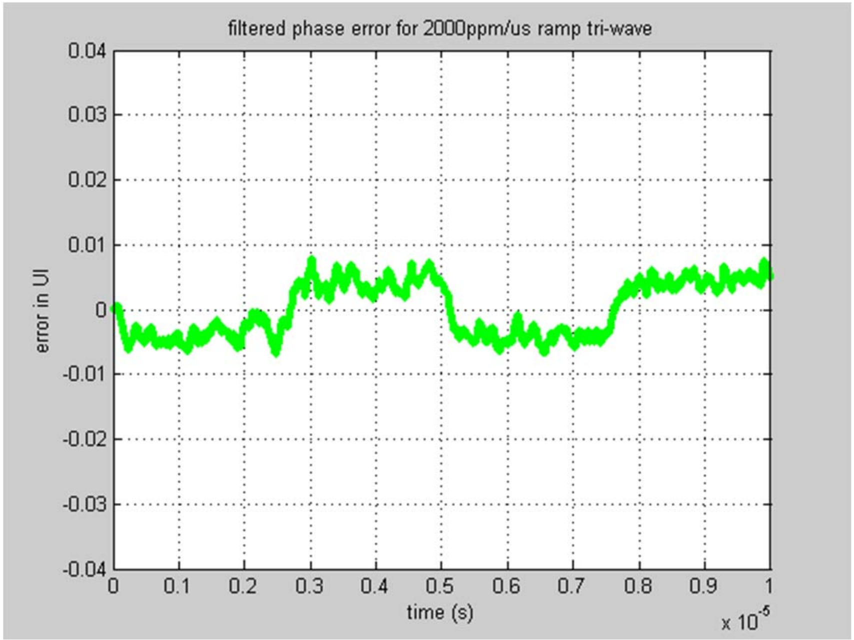
- 7000 PPM is .007UI/UI
- CDR phase integrator runs at 4UI cycle
  - Frequency intg needs to supply .028UI/cycle
- Phase integrator step is 1/256 = .0039UI
  - Freq integrator needs to be able to put in +/-7.2 per cycle (round up to 8)
  - Max PPM is 8/32/8/4*1e6 = 7812.5PPM > 7000PPM
- We need 1 sign bit and 3 magnitude bits
  - M=4
- Previous design had D_f of 7bits

PPM Slew Rate

- Want to slew frequency register at > 2000PPM/μs
  - 1μs is 5000UI at 5Gb/s, which is 312.5 frequency integrator path CDR clock cycles for frequency integrator with vote x16 decimation (312.5cycles*16UI/cycle = 5000UI)
- Want to slew 2000PPM/312.5cycles = 6.4PPM/16UI clock cycle
- PPM resolution is 1.9043 PPM from previous analysis
  - Need to shift by at more than 3.36 bits every clock cycle
  - Frequency chasing is under-determined
  - frug = 4 is bare minimum and gives us some margin

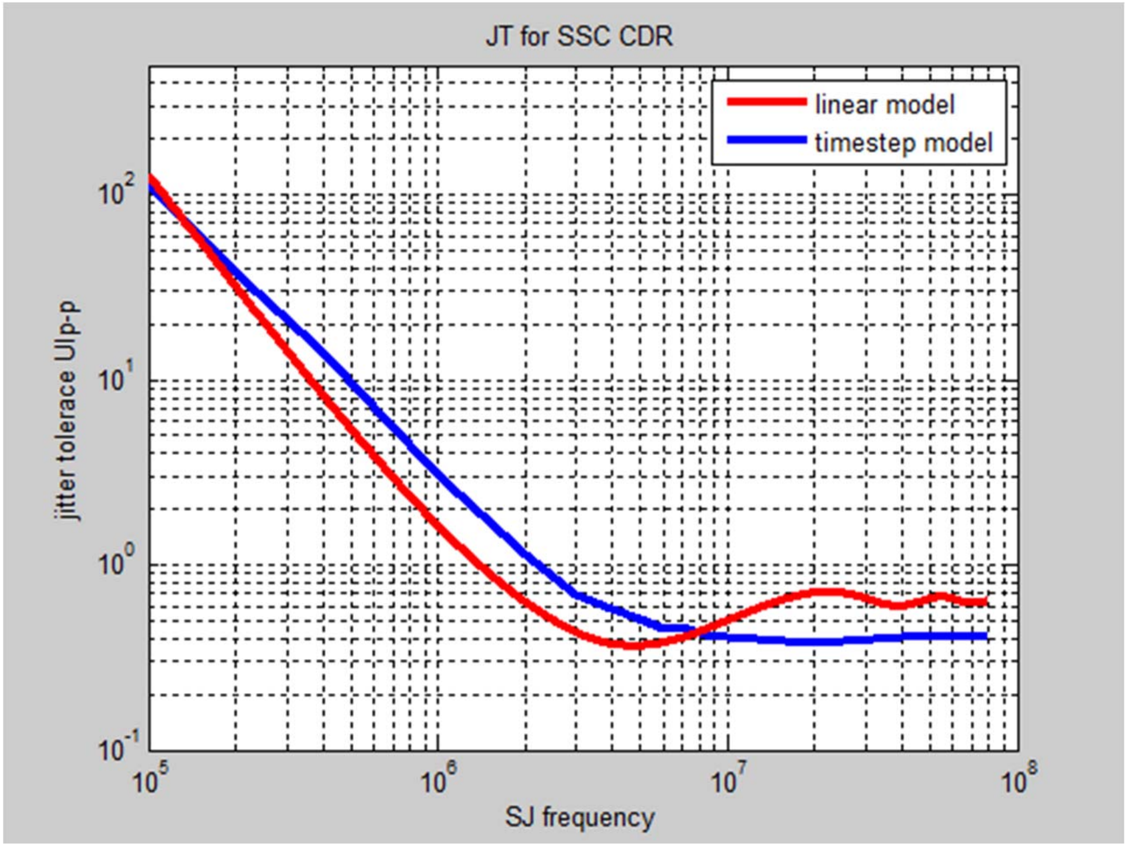
Time Step Results for SSC with 2000PPM/μs Triangle Wave





Residual error because tracking a frequency ramp with a 2nd order system is underdetermined, static error ~ .01UIp-p

Jitter Tolerance



SSC Design Summary

- Needed to increase M to allow tracking a larger PPM range
- Needed to increase integral path gain to track a faster PPM ramp
  - Increasing frug allows flexibility
- Same small signal model we started first design with that exhibited peaking - but the time step simulation results look good!
  - Jitter tolerance is a large signal event
  - Linear model is useful as an initial guide but it should not be used for final design decisions

Conclusion

- DPLL based CDR derived from an analog BB CDR
- BB phase detector modeled and studied
- CDR architecture walked through and linear model developed
- Linear model used to set initial design
- Time step simulations will always need to be done to finalize designs