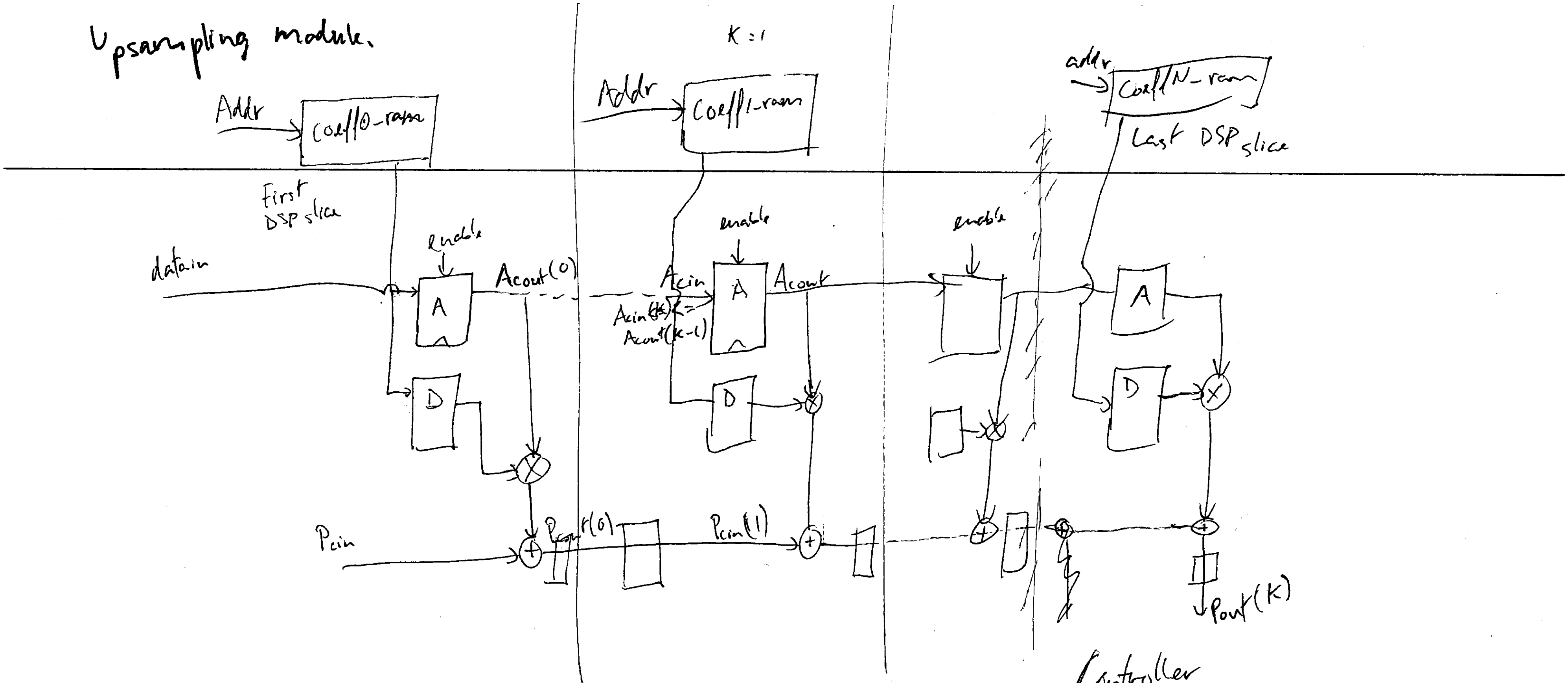
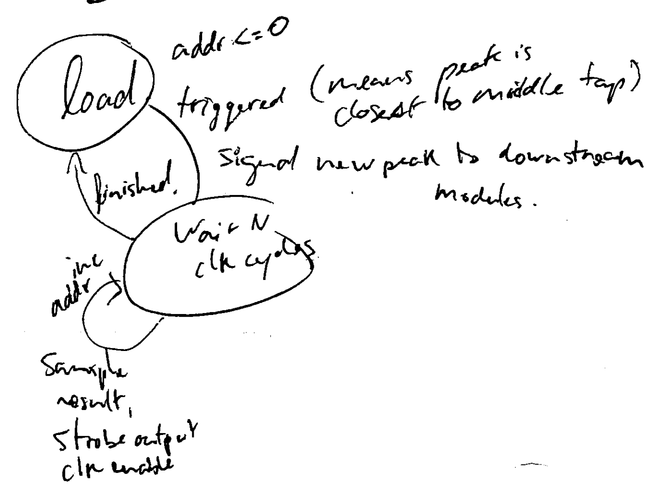
Upsampling module.



Controller



Algo: load data until it is aligned (peak in the center tap)

Set addr = 0

~~wait some clock cycles for the coeff to load~~

→ wait N clock cycles for the result to propagate

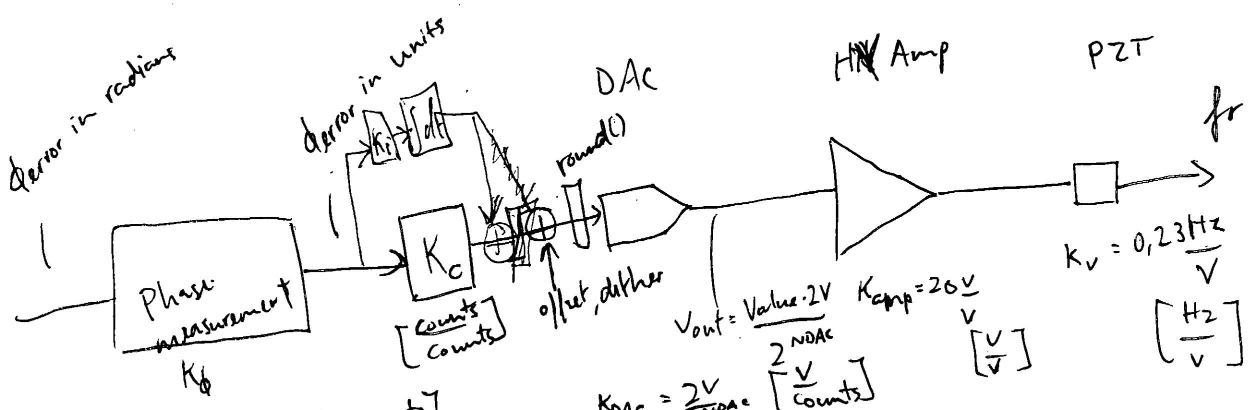
sample result at Pout(K), strobe output clockenable, inc addr.

inc addr

Processing delay for each peak: $\approx N_{upsampling}(2 + N_{inputtaps}) \approx 100 \cdot 12 = 1200$ cycles $= 6\,\mu s$

PI module

Case1: Feedback to the fast PZT



$$K_\phi = \frac{2^{48}}{2\pi} \left[\frac{Counts}{rad}\right]$$

$$K_{DAC} = \frac{2V}{2^{NDAC}} \left[\frac{V}{counts}\right]$$

NDAC = 8 (Or 7?)

Static phase error $\Delta\phi$ for a static freq. error Δf:

$\Delta f = \Delta\phi \cdot$ Overall gain $= \Delta\phi \cdot BW$

$\frac{\Delta f}{BW} = \Delta\phi$

For $\Delta f = 1\,Hz$, $BW = 200\,Hz$, $\Delta\phi = \frac{1}{200}$ rad.

Will need integral gain to reset this to 0.

Overall gain $= K_\phi K_c K_{DAC} K_{Amp} K_v$ = Close-loop BW = 200 Hz (nominal)

$$\frac{BW}{K_\phi K_{DAC} K_{Amp} K_v} = K_c = \frac{200\,Hz \cdot 2\pi \cdot 2^8}{2^{48} \cdot 2V \cdot 20\frac{V}{V} \cdot 0{,}23\frac{Hz}{V}} = \frac{2\pi \cdot 2^8 \cdot 200\,Hz}{2^{48} \cdot 2V \cdot 4{,}6\frac{Hz}{V}}$$

DAC output range: $2V \cdot 20\frac{V}{V} \cdot 0{,}23\frac{Hz}{V} = 9{,}2\,Hz$ wow... should be enough for a few seconds

Granularity of output? How much phase error do we need to make DAC input change by 1 count? 1 count of phase error

1 count of DAC input $= \frac{1}{K_c}$ Counts of phase error $= \frac{2\pi}{K_c \cdot 2^{48}}$ radians of phase error

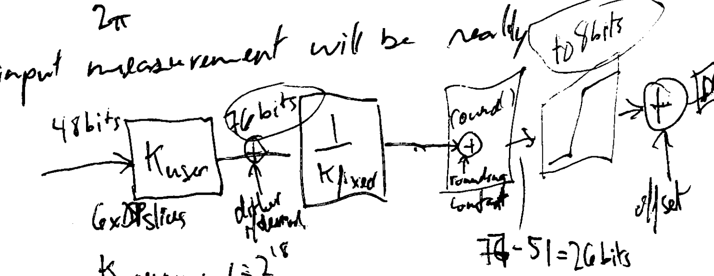
$$\boxed{\text{Nominal for 200 Hz of closed-loop BW:} \quad K_c = 124 \cdot 10^{-12} \simeq 2^{-33}}$$

Phase granularity $= 191\,\mu rad$ → time offset granularity $= \frac{191\,\mu rad}{2\pi} \cdot 5ns = 152\,fs$...

We need dither in this case because the input measurement will be really quiet and won't dither on its own.

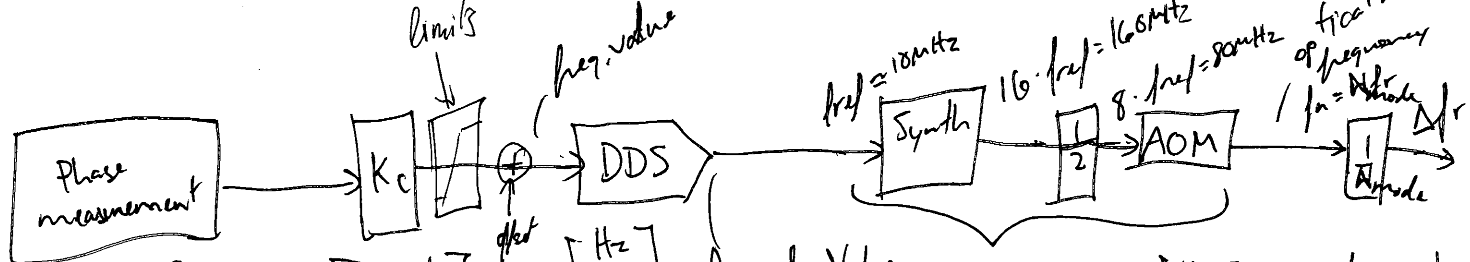
8 fractional bits at lowest possible gain, want dyn range of x1000 to $\frac{1}{1000}$ meaning 20 bits, so 28 bits total for the mult.

Need 28x48 mult. Nominal gain is $2^{10} \cdot 2^8 = 2^{18}$ then divide by $2^{-33} = \frac{1}{K_{fixed}} \cdot 2^{18} \Rightarrow K_{fixed} = 2^{18+33} = 2^{51}$



Case 2:

Feedback to the optical lock



$K_\phi = \frac{2^{48}}{2\pi}$ $\left[\frac{\text{Counts}}{\text{rad}}\right]$

$\left[\frac{\text{Counts}}{\text{Counts}}\right]$

$K_{DDS} = \left[\frac{\text{Hz}}{\text{Counts}}\right]$

$K_{DDS} = \frac{f_r}{2^{48}}$

$f_{out} = f_s \cdot \frac{\text{Value}}{2^{48}}$

$f_s = f_r$
$= f_{r,nominal} + \Delta f_r$

$K_{Doppler} = 8 \left[\frac{\text{Hz}}{\text{Hz}}\right]$

$\frac{1}{N_{mode}} \simeq \frac{1}{1e6} = 1e\text{-}6$

$$BW = K_\phi K_c K_{DDS} K_{Doppler} \cdot \frac{1}{N_{mode}}$$

$$\frac{BW \cdot N_{mode}}{K_\phi K_{DDS} K_{Doppler}} = K_c$$

$N_{mode} \simeq 1e6$

$K_\phi = \frac{2^{48}}{2\pi}$

$K_{DDS} = \frac{f_r}{2^{48}}$

$K_{Doppler} = 8$

$$K_c = \frac{BW \cdot 1e6 \cdot 2\pi}{8 \cdot f_r}$$

$= 0{,}785$ nice!

$$\Delta f_r = f_{out} \cdot \frac{K_{Doppler}}{N_{mode}}$$

$$\Delta f_r = f_s \cdot \frac{\text{Value}}{2^{48}} \cdot \frac{K_{Doppler}}{N_{mode}}$$

$$\Delta f_r = f_r \cdot \frac{\text{Value}}{2^{48}} \cdot \frac{K_{Doppler}}{N_{mode}}$$

How much ~~freq~~ shift for a $\simeq 0{,}1$ rad error?

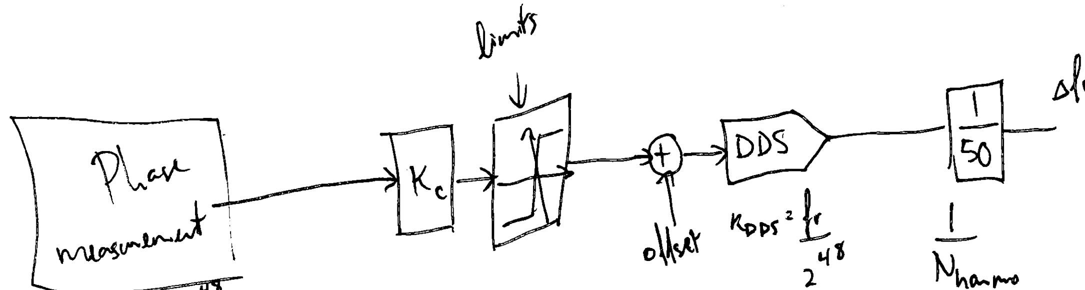
At the DDS: $0{,}1\,\text{rad} \cdot \frac{2^{48}}{2\pi} \cdot K_c \cdot \frac{f_r}{2^{48}} = \frac{0{,}1}{2\pi} \cdot K_c \cdot f_r \simeq 0{,}015 f_r \simeq 3{,}2\,MHz$

(K_c: about 1)

Will need heavy limiting!

We have limits at synth input
- AOM input

Case 3: Feedback to the DRO



$K_\phi = \frac{2^{48}}{2\pi}$

$N_{harmonic} \simeq \frac{10\,GHz}{200\,MHz} = 50$

In this case, the overall gain is:

$$K_\phi K_c K_{DDS} \cdot \frac{1}{N_{harmonic}}$$

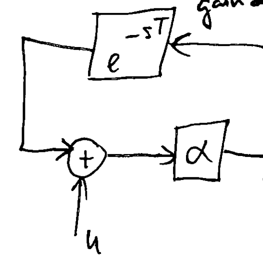
So the nominal design for BW = 200 Hz gives:

$$BW = K_\phi K_c K_{DDS} \frac{1}{N_{harmonic}}$$

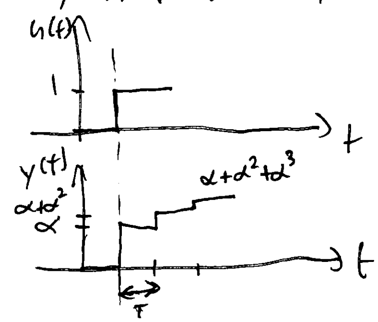
$$K_c = \frac{BW \cdot N_{harmonic}}{K_\phi K_{DDS}} = \frac{200\,Hz \cdot 50 \cdot 2\pi}{\cancel{2^{48}} \cdot \frac{f_r}{\cancel{2^{48}}}} = \frac{2\pi \cdot 50 \cdot 200\,Hz}{200\,MHz} \simeq 314 \cdot 10^{-6} \simeq 2^{-12}$$

$$K_c \simeq 2^{-12}$$

Positive feedback loop around the DDS: We are OK because loop gain α is well below 1.



If we put a unit step on u, here is what Y looks like:



So instead of having a DC gain of α, we see: $\sum_{k=1}^{\infty} \alpha^k = \sum_{k=0}^{\infty} \alpha^k - 1 = \frac{1}{1-\alpha} - 1$ $\quad |\alpha| < 1$

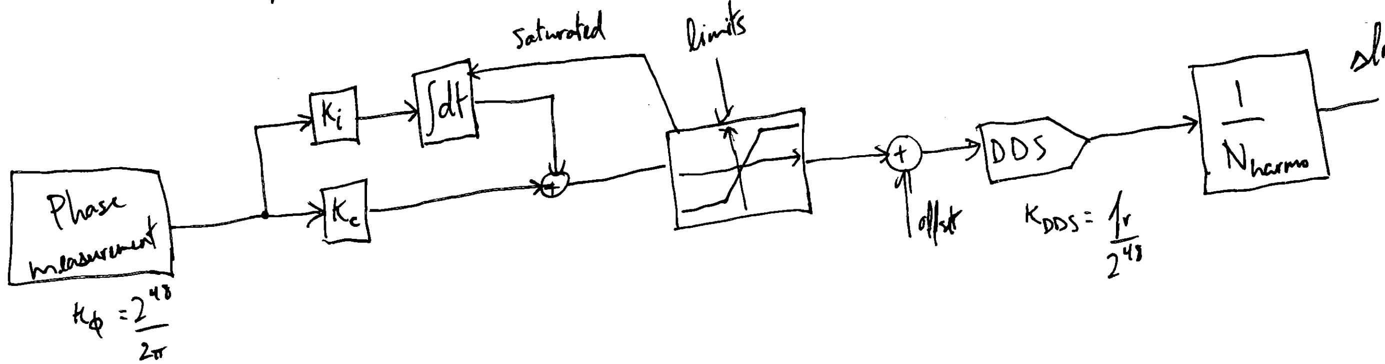
geometric series

$$= \frac{1 - 1 + \alpha}{1-\alpha}$$

$$= \alpha \cdot \frac{1}{1-\alpha}$$

which for $\alpha \ll 1$ gives $\simeq \alpha(1+\alpha)$

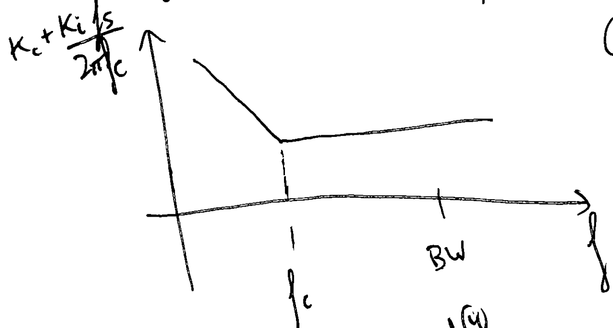
Let's figure out the integrator branch:



For K_c, we design for a given unity-gain frequency:

$$BW = \frac{K_\phi K_c K_{DDS}}{N_{harmo}} \Rightarrow \frac{BW \cdot N_{harmo}}{K_\phi K_{DDS}} = K_c \quad (1)$$

For K_i, we'll design for a specific cross-over frequency between proportional and integral gains



(2) $f_c = \beta BW$, where $\beta \simeq \frac{1}{4}$ to $\frac{1}{10}$

f_c is defined by (3) $K_c = \frac{K_i f_s}{2\pi f_c}$, where $\frac{K_i f_s}{2\pi f}$ is the integral branch's transfer function.

In this case, the integrator will be clocked at Δf_r so that the TF is:

$$|H_{integrator}(f)| = \frac{K_i \Delta f_r}{2\pi f} \quad (4)$$

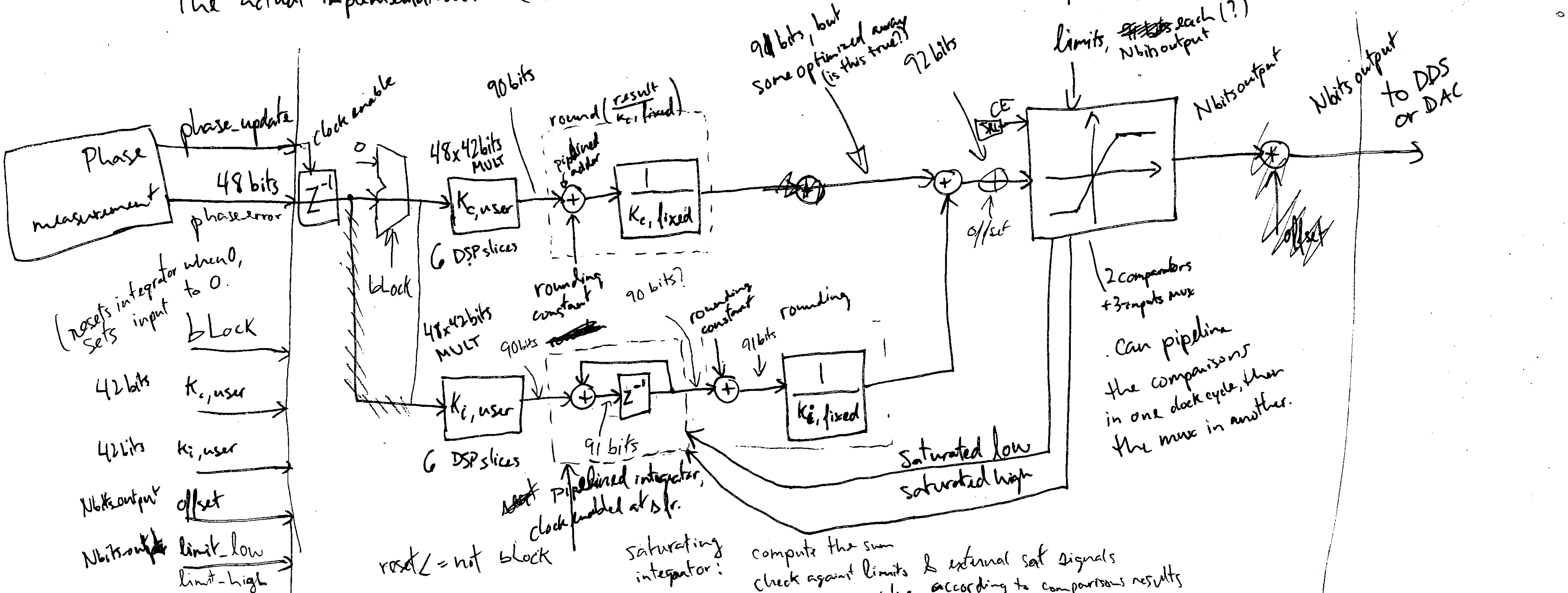
Combining (1), (2) and (3) and (4) gives: $\frac{BW \cdot N_{harmo}}{K_\phi K_{DDS}} = \frac{K_i \Delta f_r}{2\pi f_c} = \frac{K_i \Delta f_r}{2\pi \beta BW} \Rightarrow K_i = \frac{(BW)^2}{\Delta f_r} \cdot \frac{\beta \cdot 2\pi \cdot N_{harmo}}{K_\phi K_{DDS}} = \frac{2\pi\beta \cdot K_c \cdot BW}{\Delta f_r}$

Assume ~~BW=200 Hz, Δf_r = 2 kHz~~, $\beta = \frac{1}{6}$, we have $K_i \approx \frac{K_c}{10}$

$BW = \frac{\Delta f_r}{10}$

$K_i \simeq K_c \left(\frac{BW}{\Delta f_r}\right)$

The actual implementation: (covers all three cases, maybe with different fixed gains)



We choose $K_{c,fixed}$ to be some power of two so that the nominal $K_{c,user}$ is in the middle of its range minus some offset for resolution at low gains: $K_{c,user,nominal} \cdot \frac{1}{K_{c,fixed}} = K_{c,nominal}$

$K_{c,user,nominal} \approx \frac{(42\text{bits} - 8\text{bits})}{2} = 17$ bits. $K_{c,user,nominal} = 2^{17+8} = 2^{25}$

Open questions: – How many bits on the integrator signal? (Answer: need to push worst case output limits through $\frac{1}{K_{i,fixed}}$)

– Compute $K_{i,fixed}$ (see page where I give K_i as a function of K_c, ~~this~~ the same applies to $K_{i,fixed}$ vs $K_{c,fixed}$)

$K_{c,nominal}$ = Geometric mean of K_c for optical lock and DRO lock $\approx 2^{-6}$

(0,8) $\quad 2^{-12}$

$\frac{2^{25}}{2^{-6}} = K_{c,fixed} = 2^{31}$ } for DDS output

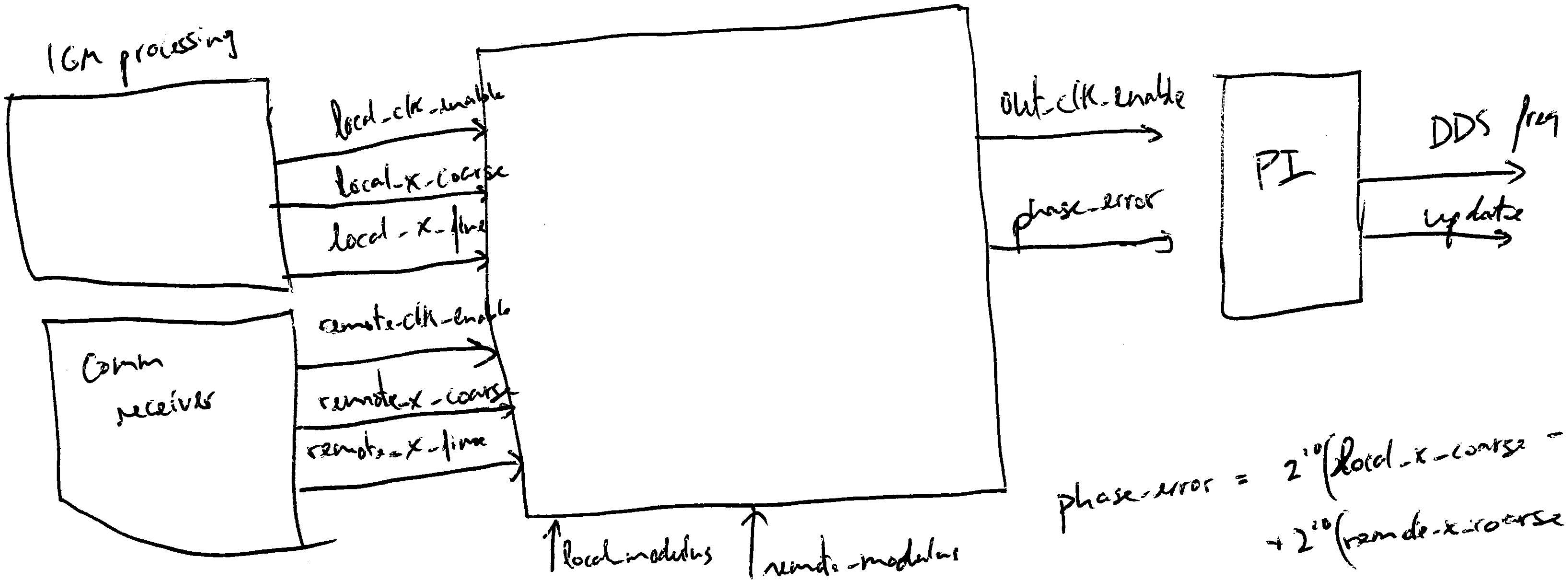
$K_{c,nominal} \approx 2^{-33}$

$\frac{2^{25}}{2^{-33}} = K_{c,fixed} \approx 2^{58}$ } for PZT output

Max integrator output required to rail the output: $\frac{intout}{K_{i,fixed}} = 2^{N_{output}-1}$

$intout = K_{i,fixed} \cdot 2^{N_{output}-1}$

Pulse_syntonization_controller_v1.vhd



$$\text{phase\_error} = 2^{10}(\text{local\_x\_coarse} - \text{local\_ref}) + \text{local\_x\_fine} + 2^{10}(\text{remote\_x\_coarse} - \text{remote\_ref}) + \text{remote\_x\_fine}$$

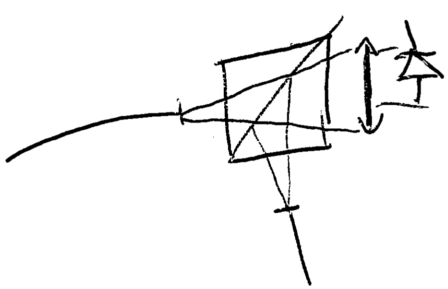
$$\Delta\phi \propto \frac{\text{timestamps}}{\text{modulus}} \cdot 2\pi$$

$$\Delta\phi = \frac{2\pi}{\text{modulus} \cdot 2^{10} \cdot 2} \cdot \text{two-way timestamps}$$

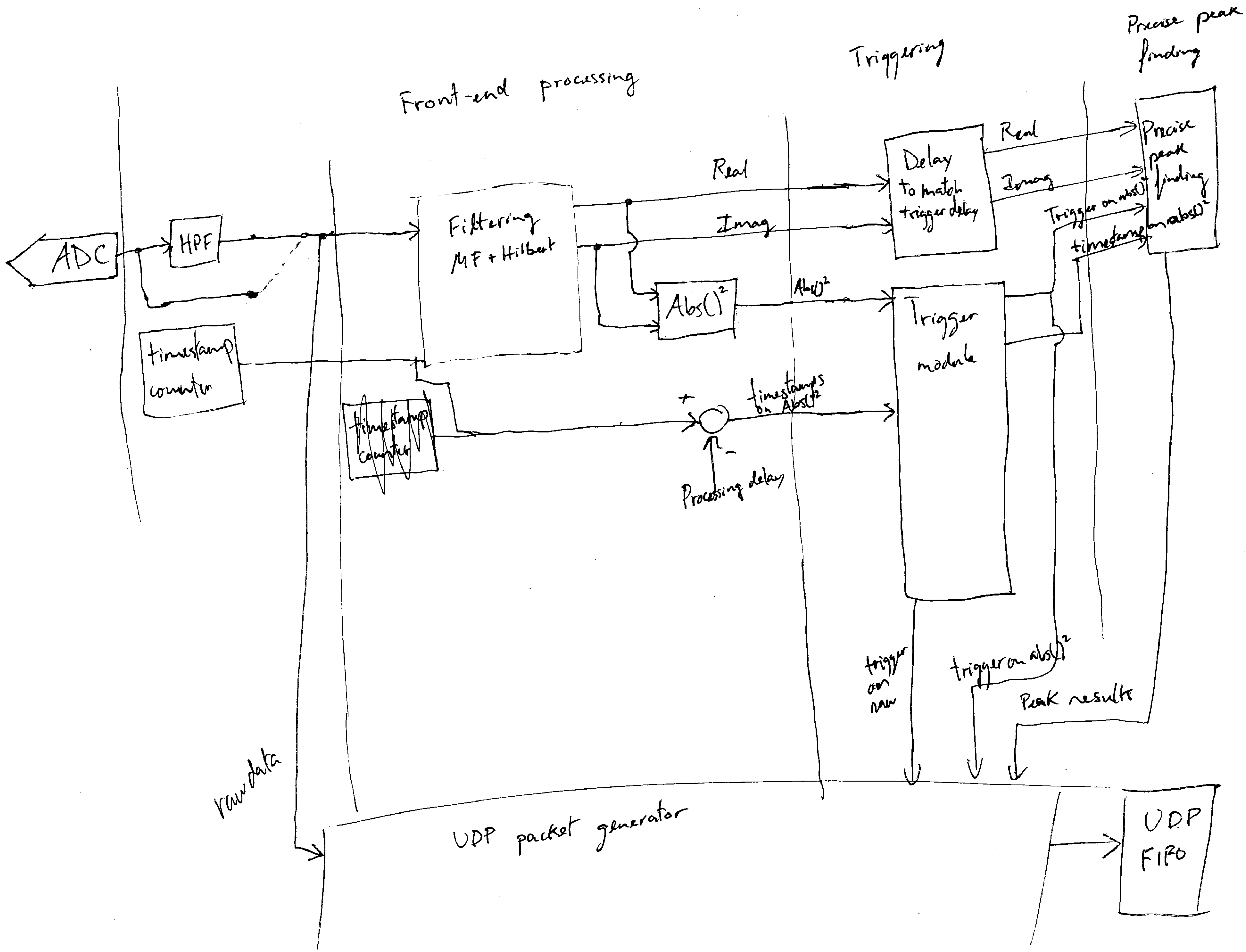
Next block expects: $\frac{2^{48}}{2\pi}\Delta\phi = \frac{2^{48}}{2^{11} \cdot \text{modulus}} \cdot \text{two-way timestamps}$

So gain is currently too low by $\frac{2^{48}}{2^{11} \cdot \text{modulus}}$

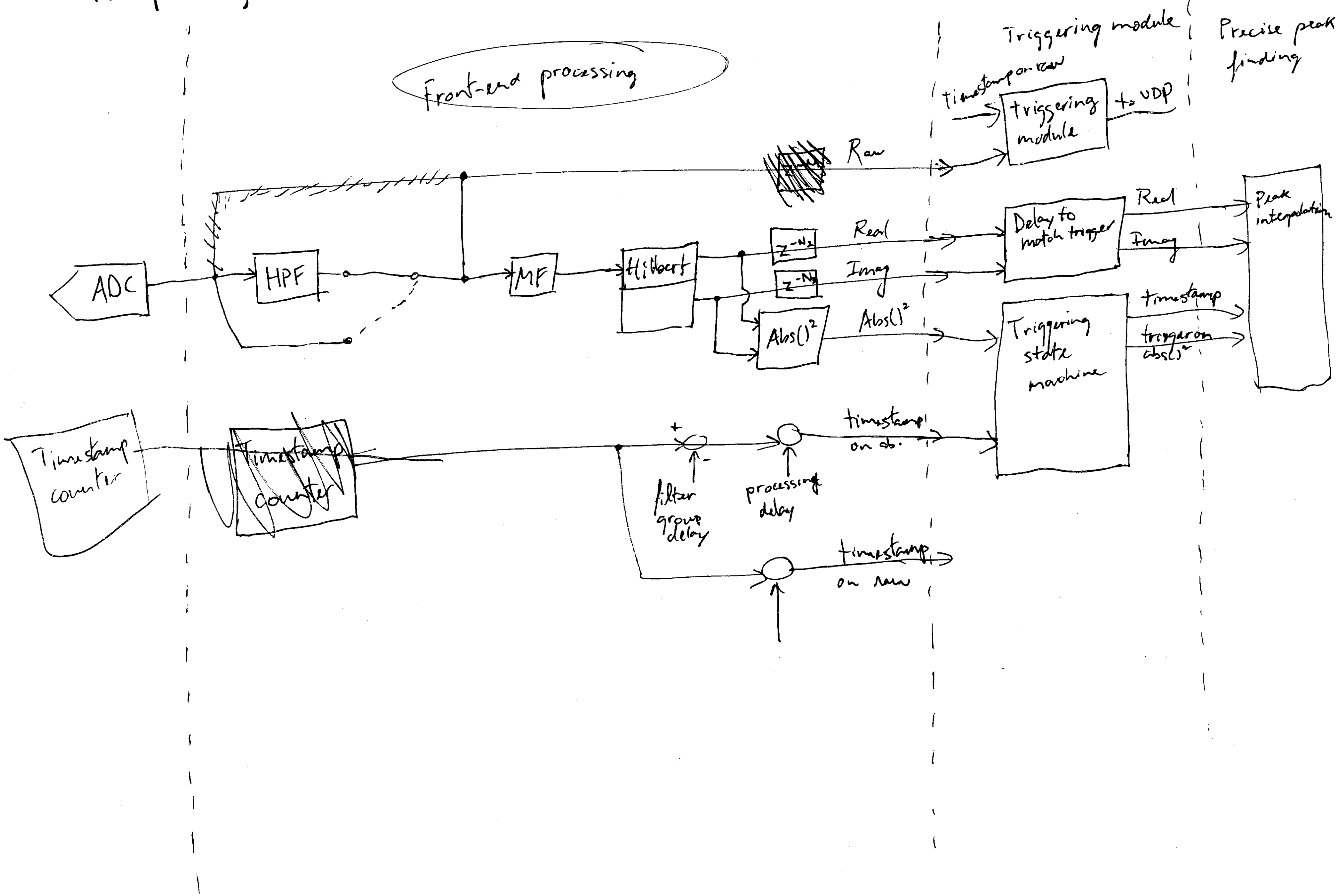
$$\simeq \frac{2^{48}}{2^{11} \cdot 100e3} \simeq \boxed{2^{20}}$$

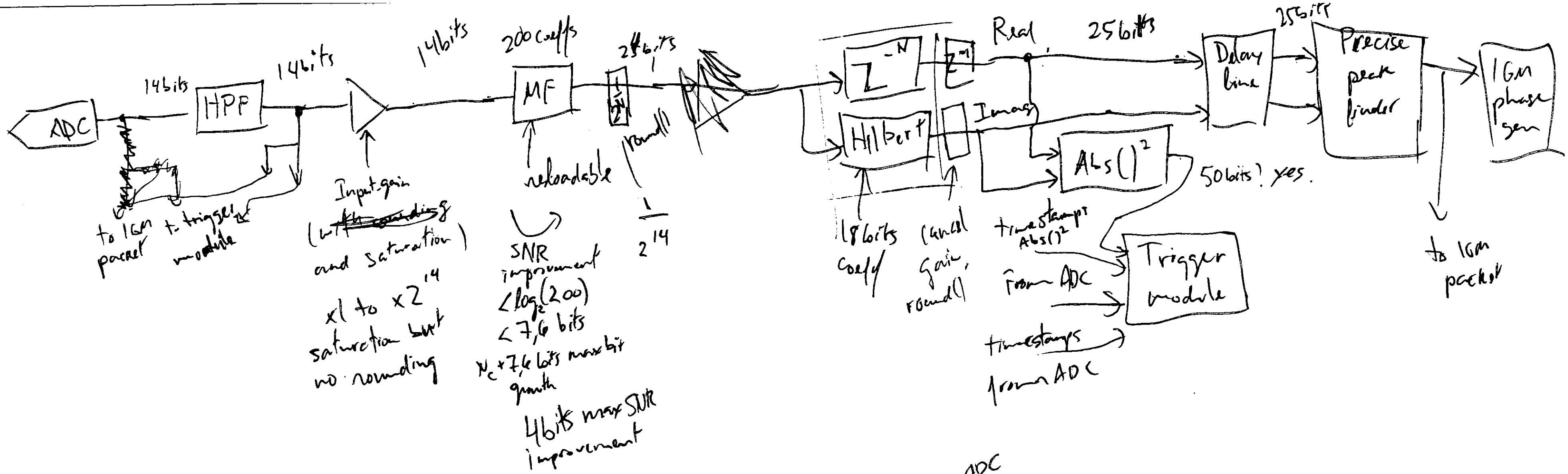


1GH processing



IGM processing





MF: 25bits coefficients.

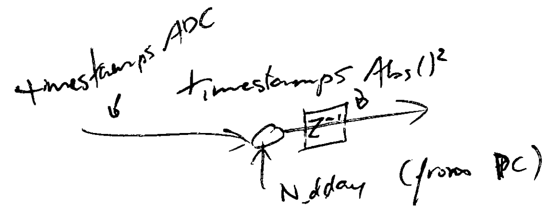
1st case: IGM already compressed.

Almost no SNR improvement.

should have no net gain.

2nd case: Long IGM: Big SNR improvement

(Best case is $\log_2 \sqrt{N} = \log_2 \sqrt{200} \simeq 3,5$ bits



Two separate modes: delay depends on trigger module only

1. Trigger on raw → UDP packets

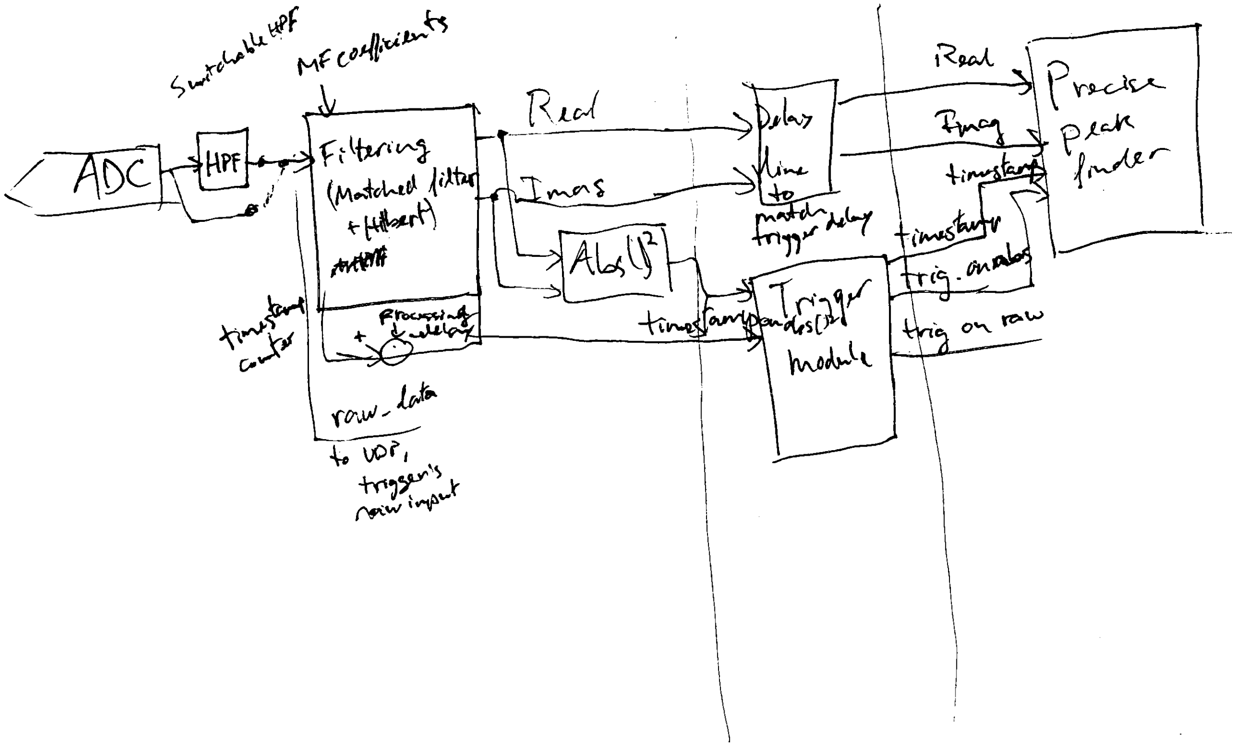
delay depends on MF's delay + Hilbert

2. Apply matched filter, trigger on abs()² → UDP packets aligned with raw peak

→ Peak finding

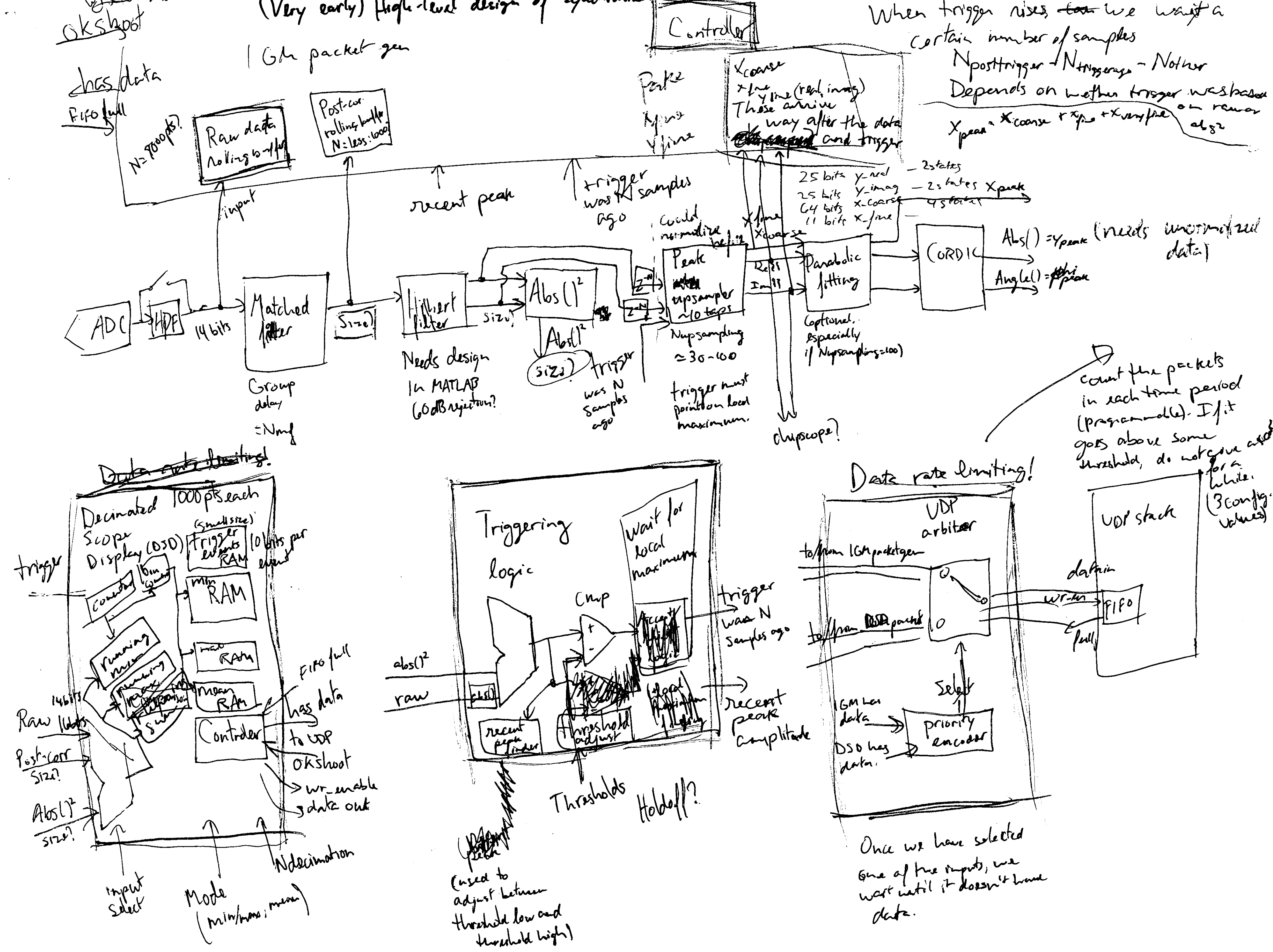
delay depends on trigger module's delay + abs()² delay

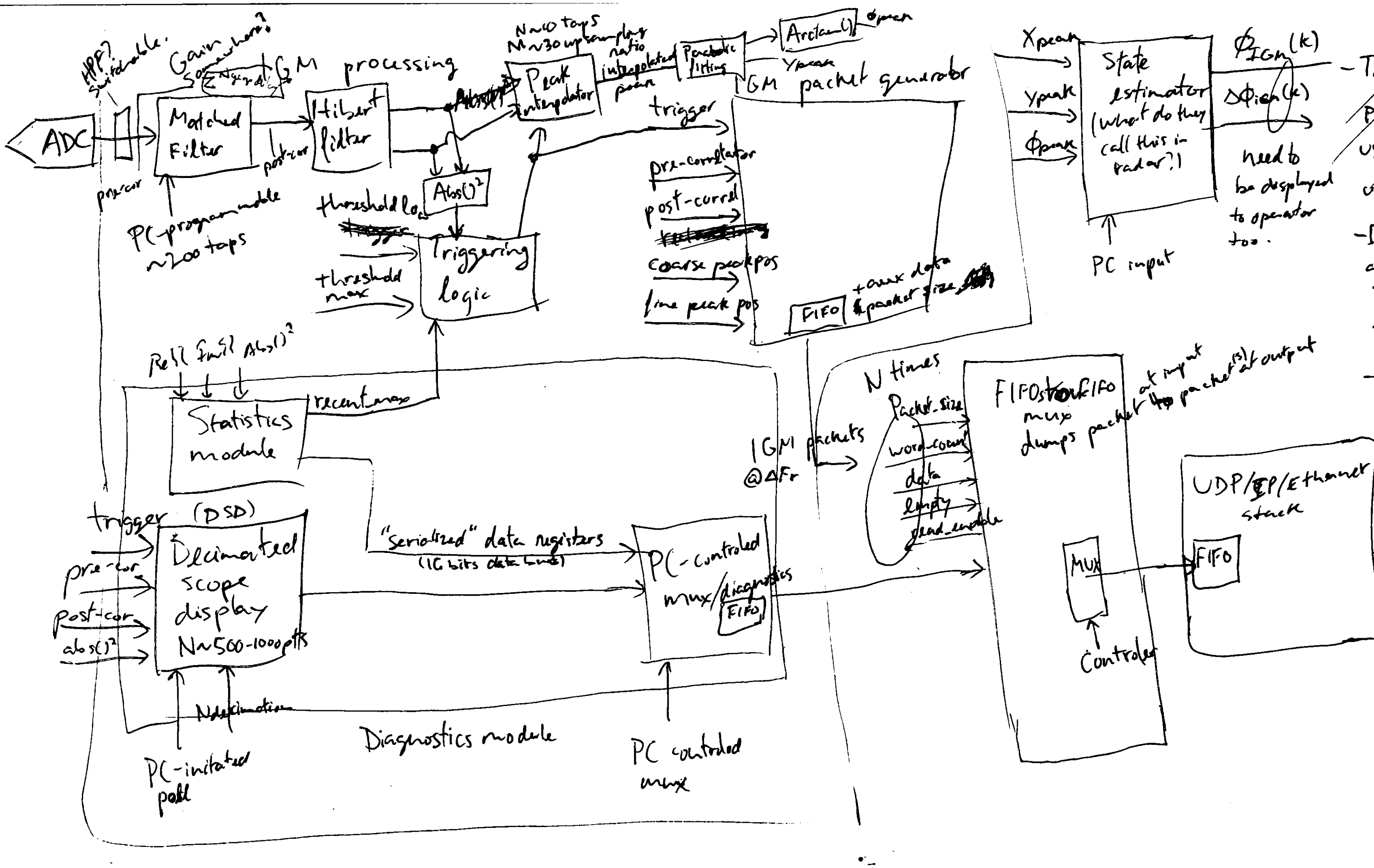
IGM processing



IGM processing

(Very early) High-level design of synchronization





- ~~To switch pre/post correlator: put the filter equal to an impulse, use real part, or have some filter, use abs()~~

- Does FIFOs to FIFO need to be aware of data rate limitations? Probably through a FIFO almost full flag from Ethernet core.

- Needs delay buffers between what ever ~~triggers~~ issues trigger and the data streams (because final decision is taken after N samples)

- Decimated scope display (DSD) also records occurrence of triggers during trace

- DSD also does mean/max per bin.

- Statistics module also computes max over last N points N is slightly higher than the nb of pts per IGM.

- DSD's main roles are:
  - Provide an easy way to read Δfr
  - Give some idea of the SNR
  - Help with setting the threshold(s)

Operation:
0. Set trigger/input source to Raw
1. DSD looks at raw input for what the user guesses is 3 periods (or a bigger number)
~~2. Statist~~ • max over each bin.
~~• mean over each bin~~

2. Statistics records max over a few ICMs, can potentially set a reasonable threshold

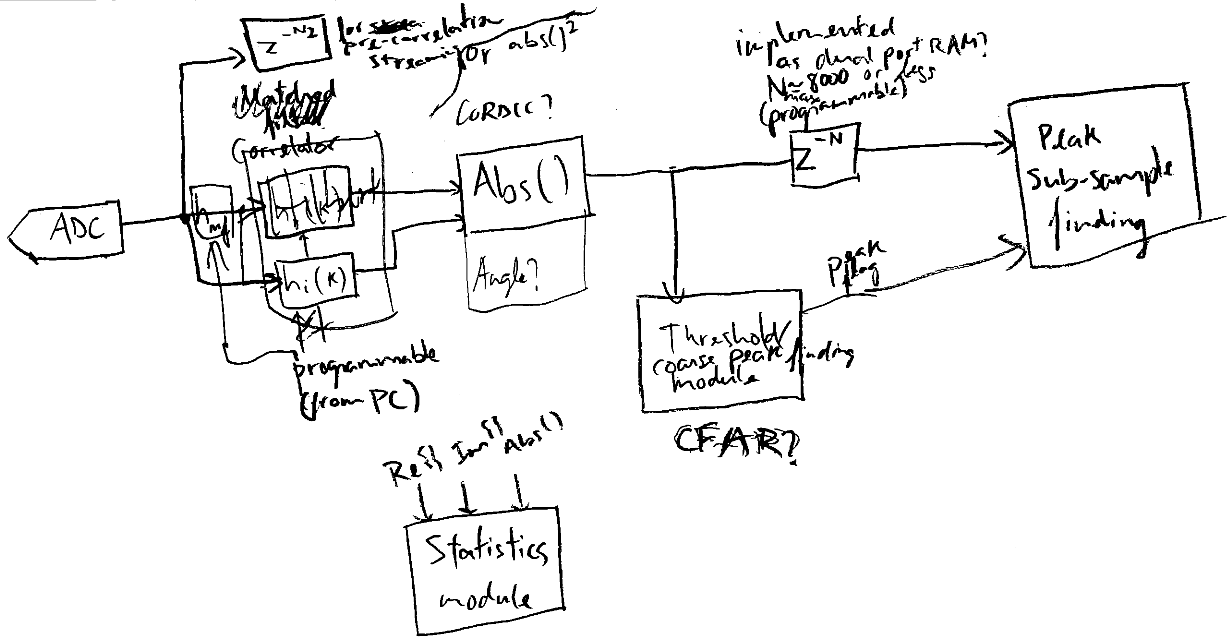
3. Stream N~100 ICMs (raw) so that PC can compute matched filter, ~~PC sends up~~

4. PC sends matched filter's impulse response, sets triggering source to $Abs()^2$, post-correlation, send ~~trigger~~ threshold levels
(set display source to post-correlation or $abs()^2$)

–

Can/should we do this without the PCs help? In high SNR conditions I think that it wouldn't be hard

Questions: What about the rest of the operation? Expected (or desired?) ΔF_r? How does this get estimated, or set by user?



Look for cross-talk in the clock buffers
→ (or use two separate)

→ Validate that the FMC104 ~~FMC108~~ and support as good timing as the NI board: Same clock, same amplitude ... tone, go through the two buffers, send the same square wave, look at level-crossings.

→ Self-reference the combs, validate no timing degradation on the IGMs (maybe a simpler test first.)

- For debug: track timetags (64 bits? mops?) for each ADC sample throughout the whole chain. will get optimized out at compile time.
- Be able to switch most block's input pre/post correlation) or disable the filter by putting just an impulse?
- IGM streaming for development mode (through Ethernet)
- Send IGMs to both ML605 and NI board to find degradation in the stability
- IGM streaming + accurate peak finding timing (pre- or post- correlation)

- Interpolator for sub-sample peak finding.

- Accept/Reject rules for the peaks? Based on (amplitude vs threshold and/or statistics) and/or timing vs expected timing?

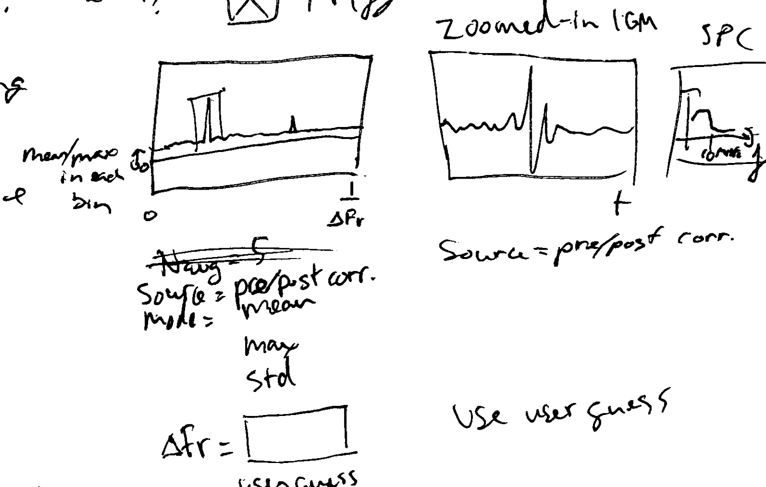
- Absolute maximum finding over N samples window, with potentially 2 overlapping windows
keep N last and output

- IGM phase ramp?
- Output to DAC?
- Knowledge of expected $\frac{1}{\Delta f_r}$? Gives robustness at low SNR, but removes robustness when Δf_r not known yet...

How about dividing the $\frac{1}{\Delta f_r}$ in N buckets (say N = 100 or N = 1000) and incoherently averaging abs() over M traces? If you know the rate well, this will work ~~extremely~~ well. Could display to the operator along with the chosen bin. If N < Number of pts between IGM, the SNR won't be that great because the peak will be "spread" over a wider range. Maybe that's not so great then. If instead we take the max ~~[illegible]~~ over each bin, then the penalty is less, and this will mimic a zoomed-out scope display (triggered at the right rate).

Will help immensely with choosing the threshold. Will also identify any parasitic IGM peak.

How about giving more than just the max ~~per~~ bin? std? mean? $\boxed{X}$ Trigger (software, just circshifts the peak in the middle)



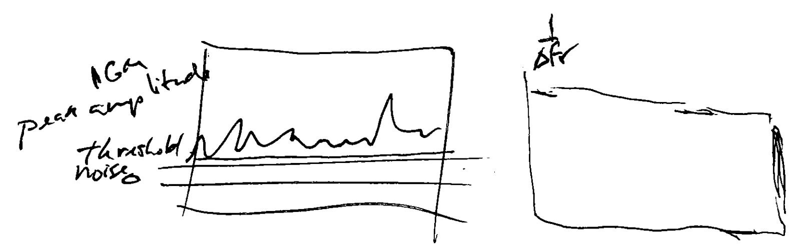
→ Having mean per bin characterizes the PDF, assuming Rayleigh (abs()) or exponential (abs()²) statistics.

→ Having std per bin allows validation of the PDF shape through the mean/std ratio.

→ How about doing this a function of IGM phase, then also having the real PDF (using a gate on the input: if above a certain threshold, ignore all N points around it) or simply through streaming of some IGM pts (continuous)

FPGA guess Δf_r = 2.944 kHz   Use FPGA guess

→ Trigger uses the last two absolute maximums to predict where the next will be (or just adjust the IGM phase axis)



ADC, Detector, and LO Noise ~~PDFs~~ PSDs through recording of short IGM traces (maybe N traces)

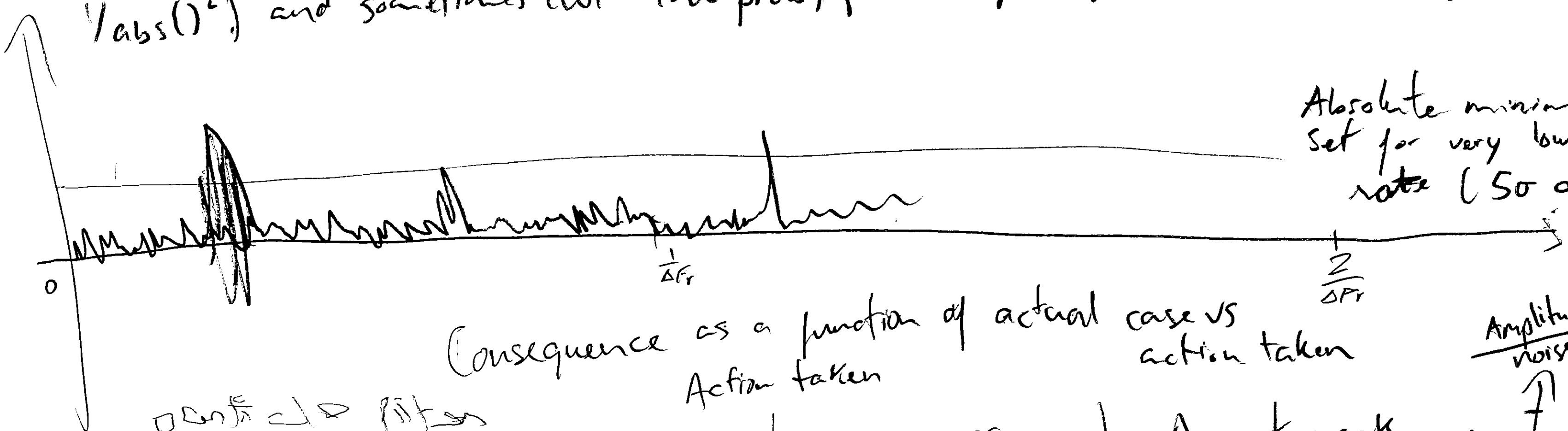
Just have a trace be the running average of the N last PSDs, then a button to switch this trace's data to a ~~[illegible]~~ new one.

I like the absolute peak finder because when it catches something, we know it's the right ~~thing~~ peak (or there's nothing, but this is solved by the absolute minimum threshold)

- Do we want a Kalman filter on the peak detection?
  + Error limiting to limit impact of false triggers

Do a literature search on Kalman filters for oscillators / peak finding / pulse detection.

Output of peak finding will be a mixture model of: gaussian time errors due to real oscillator noise (not white!) + white measurement additive noise (~~variance~~ variance scales with $1/\mathrm{abs}()^2$) and sometimes (with low prob, prob of getting this scales with $\frac{1}{\mathrm{abs}()^2}$) Uniform pdf over $\pm\frac{1}{2\Delta Fr}$



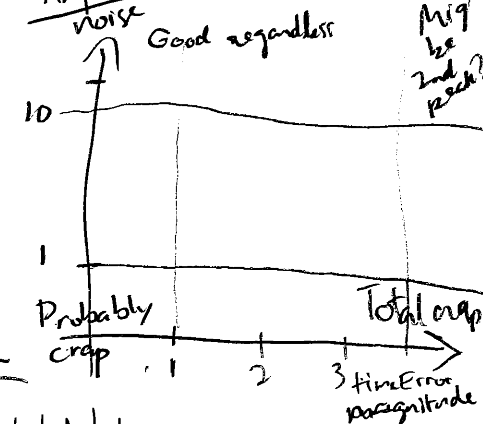
Absolute minimum threshold set for very low false alarm rate (5σ or more), helps to know pdf of noise.

Consequence as a function of actual case vs action taken

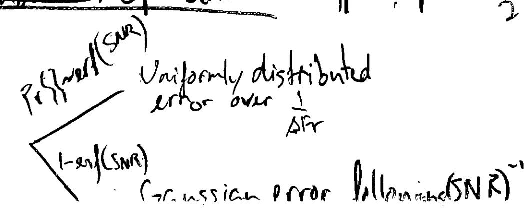
Action taken

	Reject peak	Limit error amplitude and accept	Accept peak
Wrong peak	Best choice, no consequence	Limited impact	Catastrophic
Right peak	Not using all available data	Slower convergence than possible	Best choice, nominal ~~condition~~ operation

Probability is a function of ~~P(Amplitude)~~ $\frac{\text{amplitude}}{\text{noise}}$ = SNR



a priori uncertainty



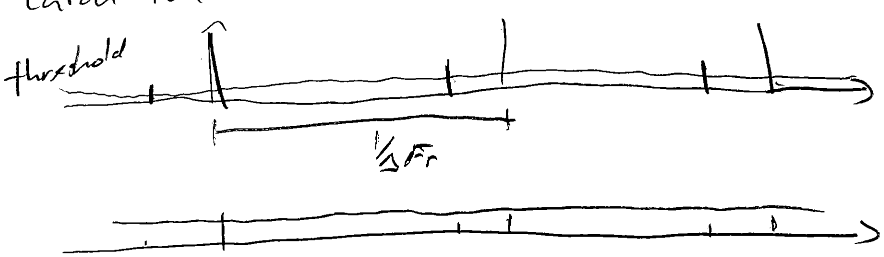
(Coarse)
Peak finding:

1. Has to be above some amplitude threshold
2. If we see higher than threshold, keep looking (to avoid triggering on a pre-main peak sidelobe) at signal to find local max in next N points.
   If you see higher: reset the N points counter
   (~~needs [illegible]~~ side delay line needs to be at least N pts long).

False positives are worst than missing IGMs: can occur anywhere in IGM, while a missing IGM should be easily detectable from expected timing... Easier with IGM phase! because it's already modulo $1/\Delta F_r$

Maybe we could have more than one threshold level:
- One absolute minimum: if below than one, assume all noise.

Hard case: Two IGMs: One big, one small, varying amplitude. It's hard not to catch the small one at high return power, while catching the big one at low return powers.



high return power: need to increase threshold

Low return power: want to decrease threshold.

→ Needs some help from either absolute peak finder or knowledge of expected timing.

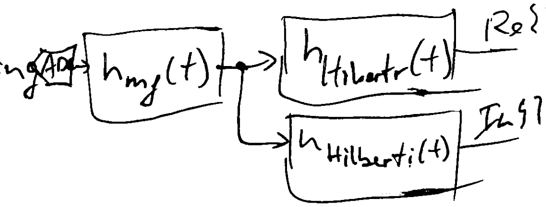
Be able to stream: N points window centered on the peak

N pts window M pts offset from the peak

N pts window anywhere (immediate trigger)

→ If correlator needs to many taps:

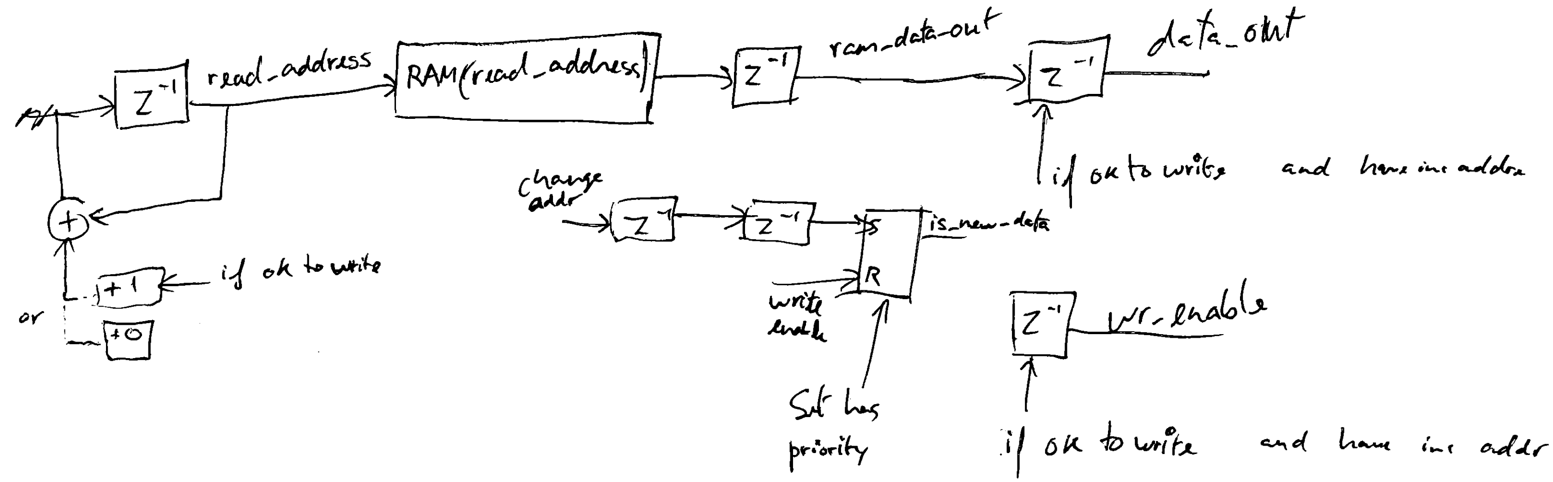
1- ~Factor of two improvement in splitting

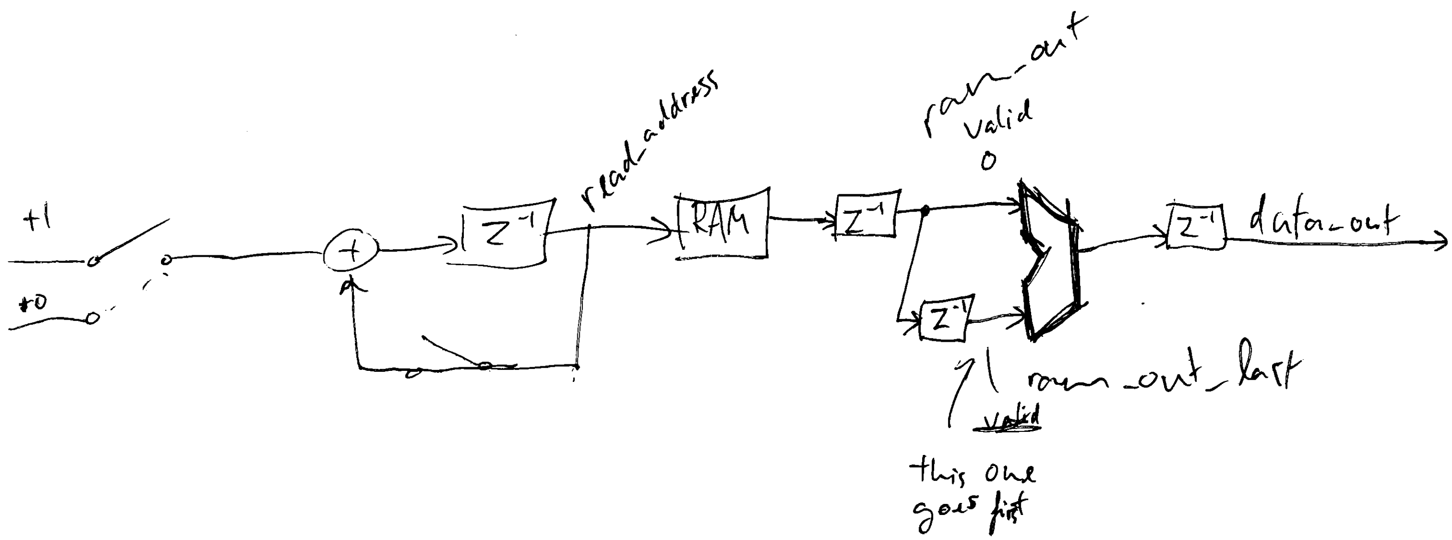


2- Optically compensate for dispersion (overall best way to do it)

3- Peak finding/phase sync, record N pts to a RAM, then compute shorter correlation will have $\frac{N^2}{M}$ cycles of delay, M being the number of MACs used in parallel.

Dumping BlockRAM to FIFO



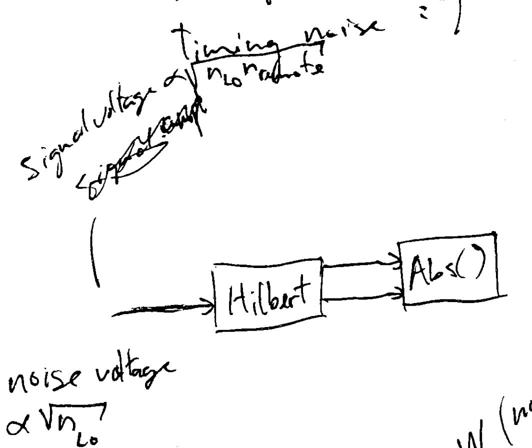


if valid(1) = '1' and fifo_full = 0 then
  outputs <= data_last
  wr_enable <= '1'
elsif valid(0) = '1' and fifo_full = 0 then
  output <= data
  wrenable <= '1'
else

What if we allow different $f_{r,LO}$, $f_{r,Remote}$? Seems very easy to see that $f_{r,LO}$ should be highest possible to match ADC, but what about $f_{r,remote}$? Assume power constrained.

SNR per IGM: ?

timing noise: ?

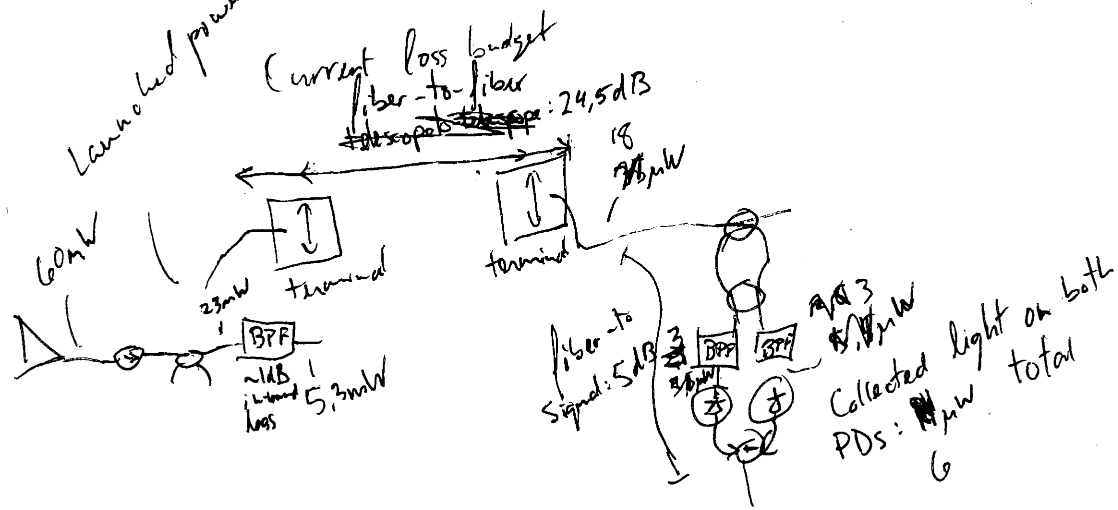


(56-2131 x3238

proposée: 14 juillet (lundi

Désirée: 16 juillet

Moq # 303-497-5980



Stuff for DARPA tutorial

SNR_{RMS} per pulse is $\sqrt{\eta n_1}$, $\quad n_1 = \dfrac{P_1}{h\nu f_r}$

$$\sigma_Y = \frac{FWHM}{SNR_{RMS}} \qquad SNR_{RMS\text{ per igm}} = SNR_{RMS\text{ per point}} \cdot \sqrt{\text{Nb of pts per IGM}}$$

$$\text{Nb of pts per igm} = \frac{FWHM}{\text{Equivalent sampling time}}$$

subject to Nyquist $FWHM \geq 2\,\text{Equiv}\,T$

$$\text{Equivalent sampling time} = \frac{\Delta f_r}{f_r^2}$$

$$\text{Nb pts per IGM} = FWHM \cdot \frac{f_r^2}{\Delta f_r}$$

$$SNR_{RMS\text{ per IGM}} = SNR_{RMS\text{ per point}} \cdot \sqrt{FWHM \cdot \frac{f_r^2}{\Delta f_r}}$$

$$\cancel{\sigma_{Y\text{ per IGM}} = FWHM\left(\sqrt{\eta n_1}\cdot\sqrt{\frac{FWHM\cdot f_r^2}{\Delta f_r}}\right)^{-1}} \quad \cancel{\text{divide by } \sqrt{\Delta f_r} = \frac{1}{\sqrt{T_{observation}}} = \frac{1}{\sqrt{\Delta f_r}}}$$

$$\cancel{\sigma_{Y\text{ per unit time}} = FWHM \cdot \sqrt{\eta n_1}\cdot\sqrt{FWHM \cdot f_r^2} = FWHM^{1.5}}$$

$$\sigma_{Y,\text{ per IGM}} = \frac{FWHM}{\sqrt{\eta n_1}\sqrt{\frac{FWHM\, f_r^2}{\Delta f}}} = \sqrt{\frac{FWHM\,\Delta f_r}{\eta n_1 f_r^2}} = \sqrt{\frac{FWHM\,\Delta f_r}{\eta n_1}}\cdot\frac{1}{f_r}$$

divide by $T_{IGM}^{1/2} = \dfrac{1}{\sqrt{\Delta f_r}}$

$$\sigma_{Y,\text{ per unit time}} = \sqrt{\frac{FWHM}{\eta n_1}}\cdot\frac{1}{f_r}$$

What are our knobs? Δf_r, f_r, P_{launch}

$$10^{15\,dB/20} = \left(\underbrace{\eta \frac{P_{received}}{E_{photon}}}_{\text{photon flux}} \cdot \underbrace{FWHM \cdot \frac{f_r}{\Delta f_r}}_{\text{Integration time}} \right)^{1/2}$$

$$10^{1.5} = \frac{P_{received}}{E_{photon}} \cdot 100/s \cdot \frac{200\,MHz}{2\,kHz} = \frac{P_{received}}{E_{photon}} \cdot 1e{-13}\,s \cdot 1e5 = \frac{P_{received}}{E_{photon}} \cdot 1e{-8}\,sec = \frac{P_{received}}{E_{photon}} \cdot 10\,ns$$

$$\frac{30\,photons}{10\,ns} = \frac{P_{received}}{E_{photon}} \Rightarrow P_{received} = 400\,pW$$

At low SNR, what breaks the system is that we need high SNR per IGM (~ 15-17 dB to have low false ~~a~~ detection rate) *

* Also conceivable to do inter-IGM averaging but much harder in practice to do coherent

$$SNR_{RMS} \text{ per IGM} = \sqrt{\eta n_1 \cdot FWHM \frac{f_r^2}{\Delta f_r}}$$ This needs to be well above 1.

$$n_1 = \frac{P_1}{f_r E_{photon}} = \frac{P_{launched}}{\eta f_r E_{photon}} \frac{1}{\text{Link loss}} = \frac{N_{launched}}{\eta \text{ Link loss}}$$

$$SNR \text{ per IGM} = \Bigg(\underbrace{\eta \frac{P_{launched}}{f_r E_{photon}} \cdot \frac{1}{\text{Link loss}}}_{\text{Received photons per pulse}} \cdot \underbrace{FWHM \frac{f_r^2}{\Delta f_r}}_{\text{Number of pulses integrated per IGM}} \Bigg)^{1/2}$$

$$= \left(\eta \frac{P_{launched}}{\text{Link loss}} \cdot \frac{1}{E_{photon}} \cdot FWHM \frac{f_r}{\Delta f_r} \right)^{1/2}$$ } These parameters all trade linearly between them.

Parameters	What drives the requirements? Drives low	Drives high
	~~- sync systematics ∝ FWHM~~ ~~- Systematics synth ∝ FWHM overall sensitivity (timing noise floor) ∝ FWHM~~	SNR per IGM $\propto (FWHM)^{1/2}$
FWHM	- Processing effort per IGM	
Δf_r (1 Hz, to 100 kHz)	$\underbrace{\text{- Acquisition time; - Link fluctuations sampling rate (turbulence, Doppler); - Clock error sampling rate}}_{\text{drives high}}$ vs	$\underbrace{\text{SNR per IGM} \propto (1/\Delta f_r)^{1/2}\text{; Sync systematics} \propto \Delta f_r}_{\text{drives low}}$
$P_{launched}$	$\underbrace{\text{- Practical limits (amplifiers, nonlinearity, safety)}}_{\text{drives low}}$ vs	$\underbrace{\text{SNR \{ Go/No-go; - timing noise floor}}_{\text{drives high}}$
f_r	$\underbrace{\text{- ADC rates; - fceo generation}}_{\text{drives low}}$ } vs	$\underbrace{\text{SNR}}_{\text{drives high}}$

Other constraints: Nyquist: $FWHM \cdot \frac{f_r^2}{\Delta f_r} > 2$

Put some numbers in.

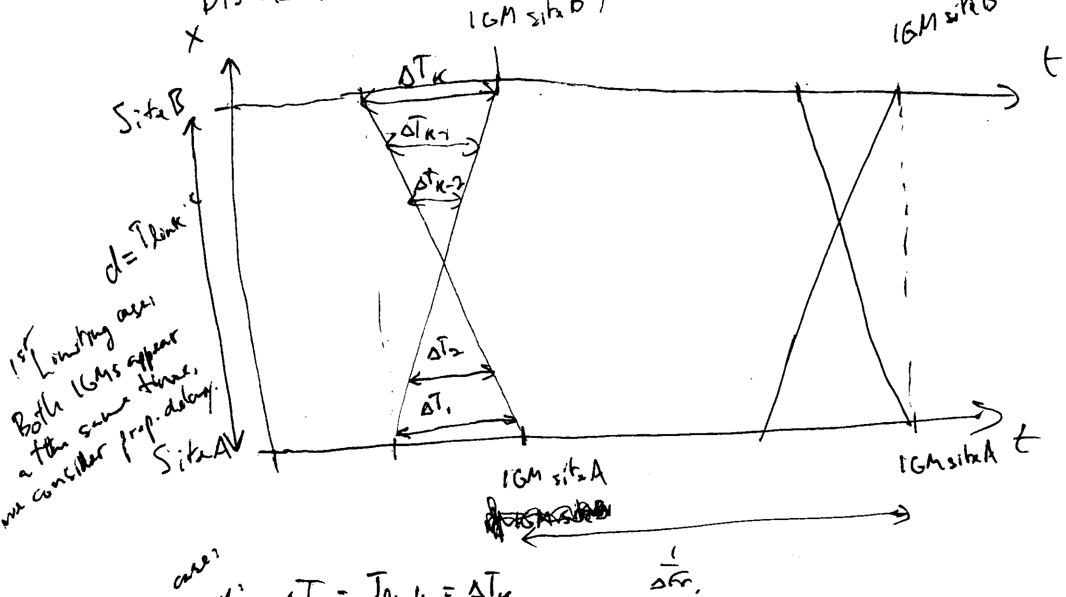
$\underbrace{FWHM \cdot \frac{f_r^2}{\Delta f_r}}$ Nb pts per IGM

→ If significant unknown Doppler and, search space becomes 2D (delay-freq) instead of 1D (delay)

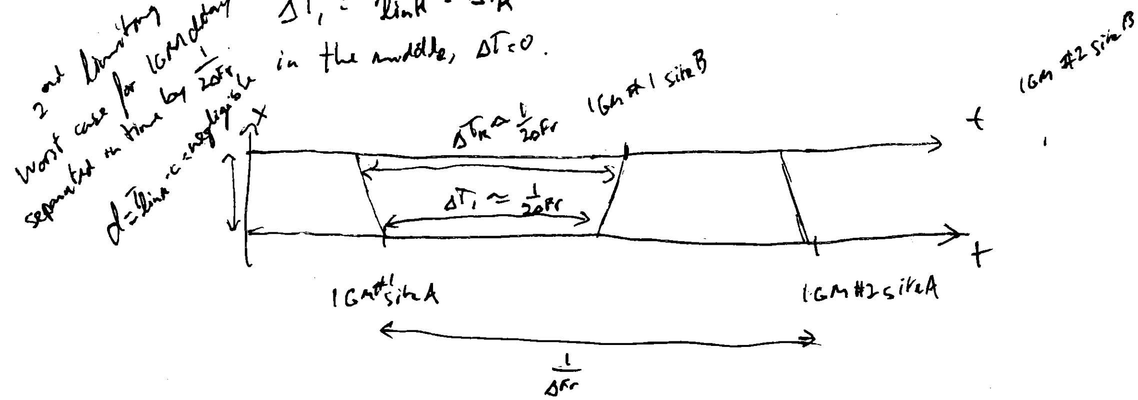
this means: - False alarm rate worse (need more SNR margin b/c more degrees of freedom)
- Significant computational overhead.

or: incoherent integration.

Distributed-delay model of turbulence piston effect



$\Delta T_1 = T_{link} = \Delta T_K$ in the middle, $\Delta T \simeq 0$.

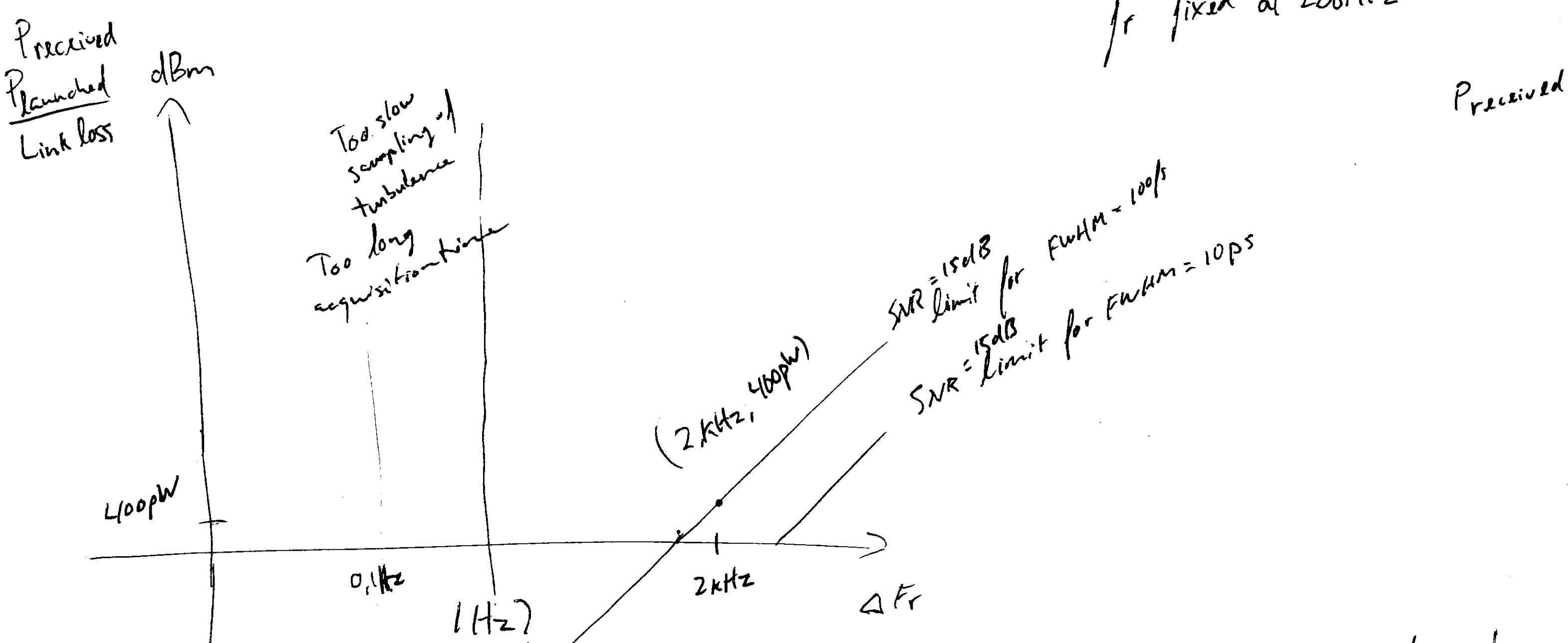


The ~~peak~~ layers of air at different distance get sampled at different delays. ΔT_1, ΔT_2, etc.

This case corresponds to IGM appearing at site A and B at the same time. Neglecting the propagation delay T_{link}, the two IGMs could appear as much as $\frac{1}{2\Delta f_r}$ apart in time. In this case all air layers get sampled with approximately the same delay $\frac{1}{2\Delta f_r}$.

If $d = 100$ km, $T_{link} = 333\,\mu s$, which is equivalent to a delay of $\frac{1}{2\Delta f_r} = 333\,\mu s \Rightarrow \Delta f_r = 1{,}5$ kHz, which is quite fast. So the IGM delay will almost always dominate the propagation delay (at best they are of same scale).

f_r fixed at 200MHz



How quickly do we have to sample the turbulence?

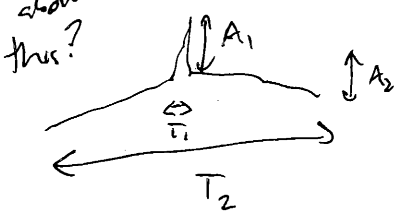
We make a diff of the turbulence with a worst case delay of $\frac{1}{\Delta f_r} \cdot \frac{1}{2}$

For a given PSD, how quickly do we need to sample to hit

Turbulence PSD is $K f^{-8/3} \left[\frac{s^2}{Hz}\right]$ I think?

$K f^{-2.3}$.

When diff'ed, it becomes $K f^{-0.3}$, almost white timing noise. How about sending something like this?



How much is $\int_{f_{min}}^{f_{high}} S(f) |H_{meas}(f)|^2 df$

$H_{meas}(f) = \mathcal{F}\{\delta(t) - \delta(t-T)\}$

$\exp(j2\pi f \cdot 0) - \exp(j2\pi f \cdot T)$ Decouples acquisition time from precision.

$= 2\sin\left(\frac{2\pi f T}{2}\right) \cdot \phi(f)$

$\sigma^2_{turb}(T) =$

$\sigma^2_{turb}(T, f_{min}, f_{high}) = \int^{f_{high}} S(f)\, 4\sin^2(\pi f T)\, df$

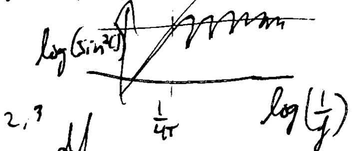
$= \int_{f_{low}}^{f_{high}} K f^{-2.3}\, 4\sin^2(\pi f T)\, df$

Can also approximate $H_{meas}(f)$ by just αf^2 up to some cutoff

$K \simeq 10^{-29} \frac{s^2}{Hz}$ (Laurent)

$$\sigma^2_{turb}(T, f_{min}, f_{max}) = 4K \int_{f_{min}}^{f_{max}} f^{-2,3} \sin^2(\pi f T)\, df = 4K \cdot I(T, f_{min}, f_{max})$$

we approximate $\sin^2(\pi f T)$ in two parts



$$I(T, f_{min}, f_{max}) \simeq \int_{f_{min}}^{f_{max}} f^{-2,3} \sin^2(\pi f T)\, df \simeq \int_{f_{min}}^{\frac{1}{4T}} f^{-2,3} (\pi f T)^2\, df + \int_{\frac{1}{4T}}^{f_{max}} f^{-2,3}\, df$$

$(\pi f T)^2 = 1$

(upper bound)
$$\simeq \int_{f_{min}}^{f_m} f^{-0,3} \pi^2 T^2\, df + \int_{f_m}^{f_{max}} f^{-2,3}\, df \simeq \pi^2 T^2 \int_{f_{min}}^{f_m} f^{-0,3}\, df + \int_{f_m}^{f_{max}} f^{-2,3}\, df$$

$f_m^2 = \frac{1}{(\pi T)^2}$

$f_m = \frac{1}{\pi T}$

$$\simeq \cancel{\frac{1}{-2,3+1} f^{-1,3}\Big|_{f_{min}}^{f_m}} = \frac{1}{-1,3}\left(f_{max}^{-1,3} - f_m^{-1,3}\right) + \frac{\pi^2 T^2}{0,7}\left(f_m^{0,7} - f_{min}^{0,7}\right)$$

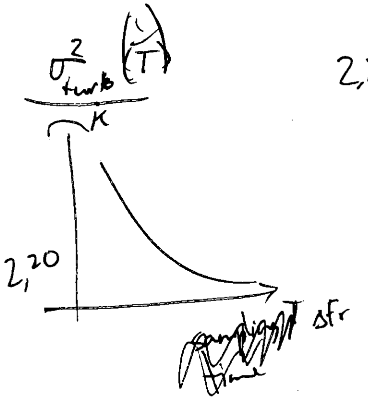
lim as $f_{min} \to 0$?

$$\simeq \frac{1}{-1,3}\left(f_{max}^{-1,3} - f_m^{-1,3}\right) + \frac{\pi^2 T^2}{0,7} f_m^{0,7}$$

lim as $f_{max} \to \infty$?

$$\simeq \frac{f_m^{-1,3}}{1,3} + \frac{\pi^2 T^2}{0,7} f_m^{0,7} \qquad f_m = \frac{1}{\pi T}$$

$K \sim 10^{-29} \frac{s^2}{Hz}$



$2,2(\pi T)^{1,3} \simeq 4e{-4}$ Hz at $\Delta f = 1$ kHz

$\simeq 4$ Hz at $\Delta f = 1$ Hz

$$\sigma^2_{turb}(T) \simeq 4K\left[\frac{(\pi T)^{1,3}}{1,3} + \frac{1}{0,7(\pi T)^{0,7}}\right]$$

$$\simeq 4K\left[\underbrace{\frac{(\pi T)^{1,3}}{1,3}}_{\text{low-freq contrib.}} + \underbrace{\frac{(\pi T)^{1,3}}{0,7}}_{\text{high freq contrib}}\right] = 4K(\pi T)^{1,3} \cdot 2,20$$

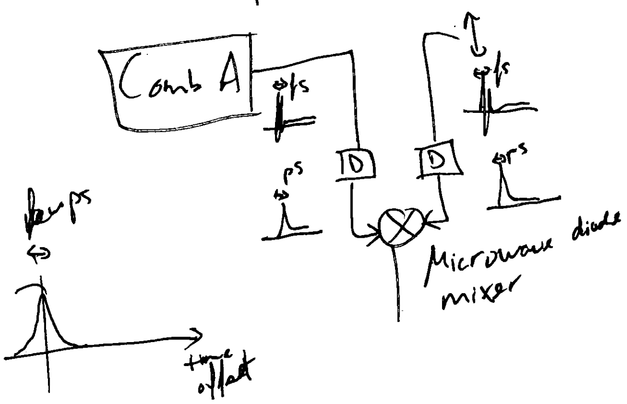
low-freq contrib. higher T means higher slope from $\sin^2(\pi f T)^2$

~~low~~ high freq contrib. lower T means lower freq integration limit.

This is a "lumped-delay" model. ~~Closer~~ Better model would be based on many small delays, all having timing PSD $\frac{S(f)}{\Delta L}$, sampled at different differential delays.

$$\int_{\Delta L} \frac{S(f)}{\Delta L} |h_{meas}(f, L)|^2$$

Microwave pulse comparison:

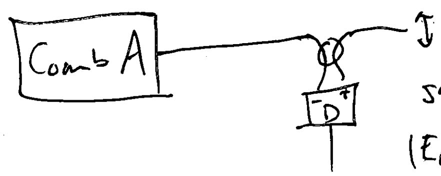


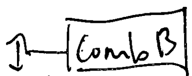
"Comparing the time of the current pulses"
Measures current pulses overlap

Delay of PD must be stable.

- Sensitivity of PD is critical
- Response time stability

Optical pulse comparison:



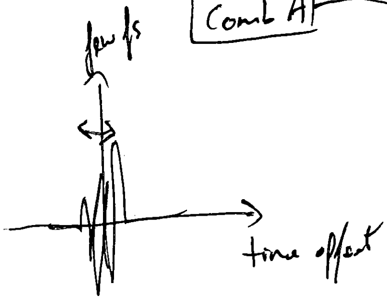


square-law ~~detector~~ balanced detector:

$|E_1|^2 + |E_2|^2 + 2|E_1E_2| - \left(|E_1|^2 + |E_2|^2 - 2|E_1E_2|\right) = 4|E_1E_2|$

Measures ~~field~~ electric field overlap

$\equiv$



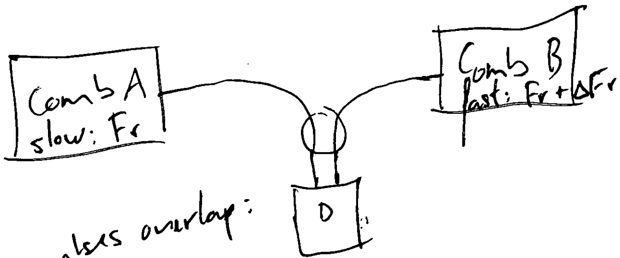
"Optical field product"

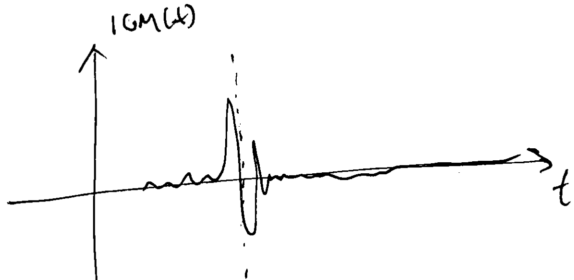
- Sensitivity of PD not as important: coherent gain from LO.
- Delay of the PD is non-critical: only delays the information of their overlap

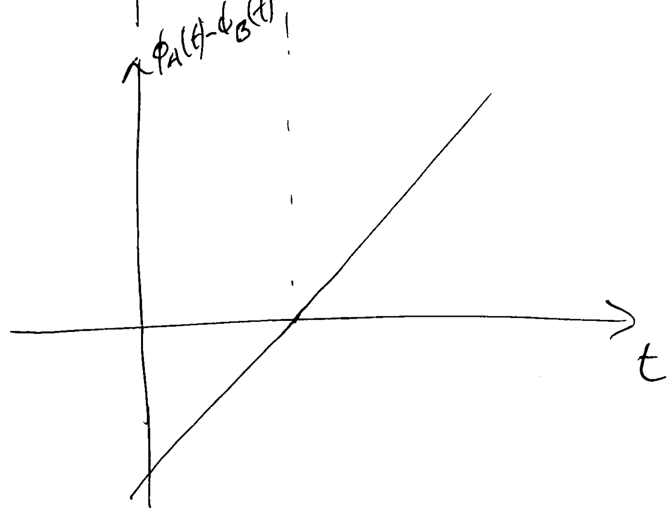
Fiber delays become the limiting factor:

1m of fiber: $50\,\text{fs}/{}^\circ\text{C}$

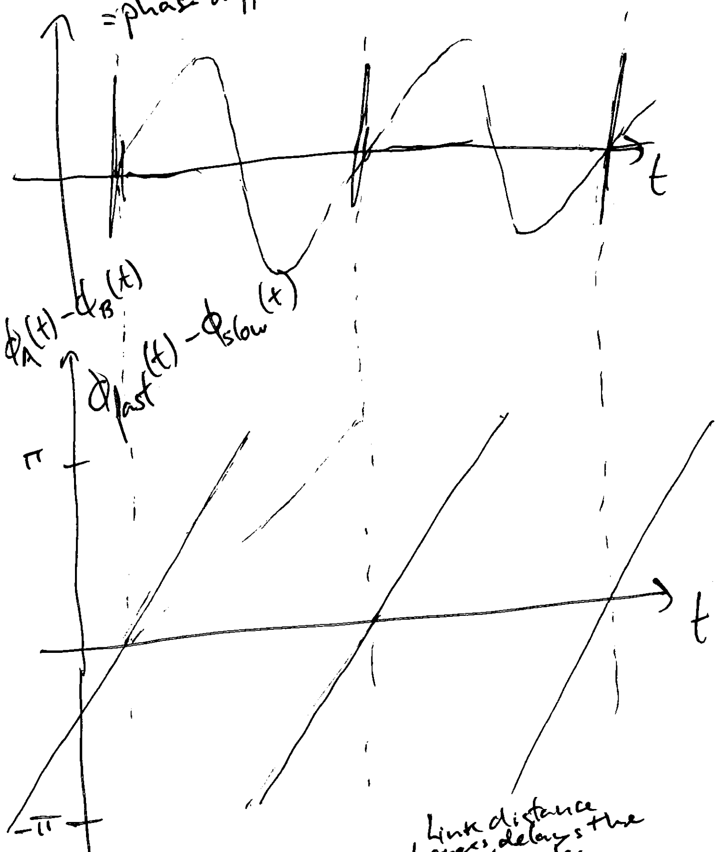
No link, single site:





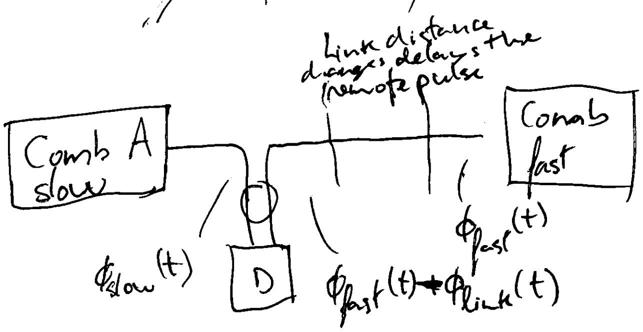


When pulses overlap: interference = phase difference = 0



→ Every IGM ~~updates~~ gives a highly precise update of the phase difference of the two pulse trains.

→



IGM phase sees pertubed remote phase

$$\phi_{igm}(t) = \phi_{fast}(t) - \phi_{link}(t) - \phi_{slow}(t)$$

Solution: two-way

Slide 2: Caveats: No information when pulses don't overlap.

~~Thus~~ → We need to continuously scan the delay to get phase difference if it's anything other than 0.

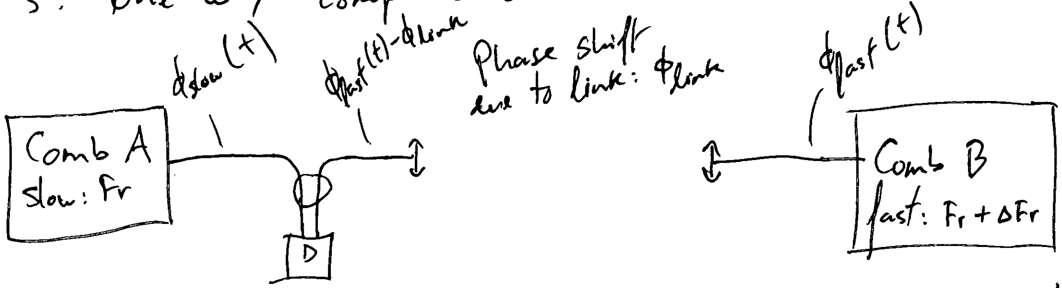
~~Hence the ss~~

Most elegantly solved by slight rep rate detuning.

Same cartoon, this time with f_r and Δf_r.

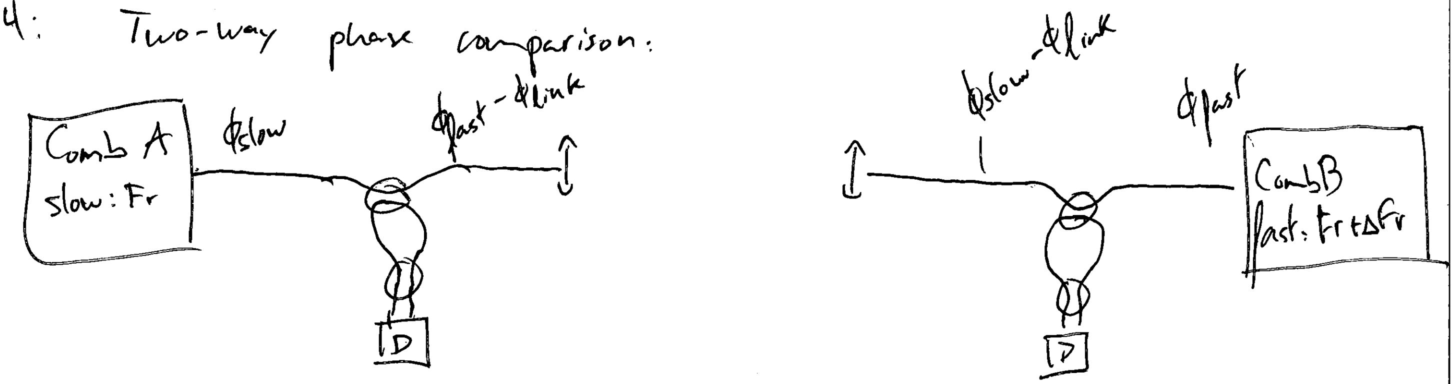
→ Phase offset due to ~~path~~ delay

Slide 3: One-way comparison:



$$\phi_{IGM} = (\phi_{fast} - \phi_{link}) - \phi_{slow}$$ → Sensitive to path length change due to turbulence/motion

Slide 4: Two-way phase comparison:



$$\phi_{IGMA}(t) = \phi_{fast} - \phi_{link} - \phi_{slow}$$

$$\phi_{fast} - \phi_{slow} - \phi_{link}$$

$$\phi_{IGMB}(t) = \phi_{fast} - (\phi_{slow} - \phi_{link})$$
$$= \phi_{fast} - \phi_{slow} + \phi_{link}$$

Data link
~~Exchange~~ data through comm link

$$\phi_{IGMA} + \phi_{IGMB} = 2(\phi_{fast} - \phi_{slow})$$

~~Closing the loop on this signal phase-locks~~

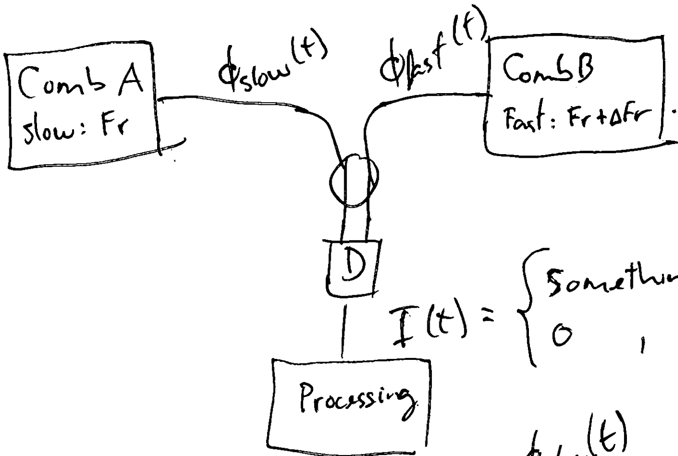
Forcing this signal to be a perfect ramp means that the two sites have an exact constant frequency offset.

"Syntonization" (?) Even though they have two different frequency b/c this freq diff is known.

Slide 1: Dual-comb LOS ~~Ms~~ can be viewed as a technique which makes a very precise phase comparison ~~technique~~ between two pulse trains.

First slide without ΔFr please

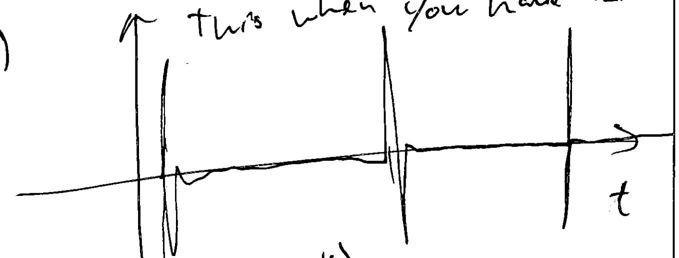
Just $\phi_A(t)$ and $\phi_B(t)$

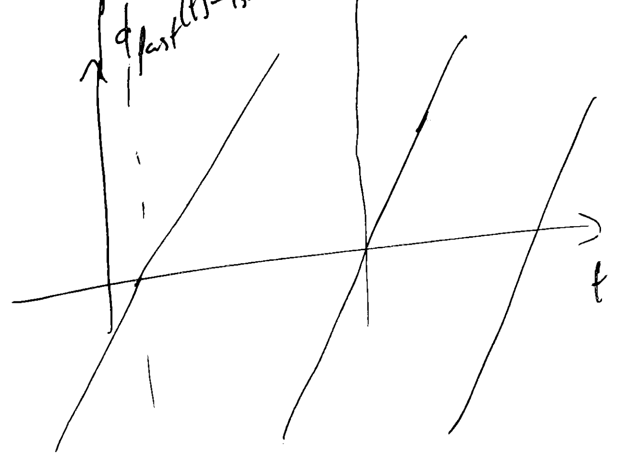


$$\phi_{IGM}(t) = \phi_{fast}(t) - \phi_{slow}(t)$$

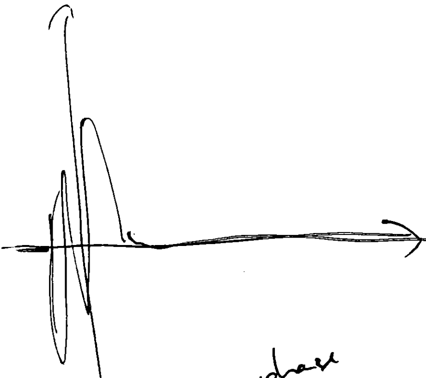
$$I(t) = \begin{cases} \text{Something} & \text{if } \phi_{fast}(t) - \phi_{slow}(t) < 2\pi \cdot 100fs \cdot Fr \\ 0, & \text{if } \phi_{fast}(t) - \phi_{slow}(t) > 100fs \end{cases}$$

This when you have a ΔFr.

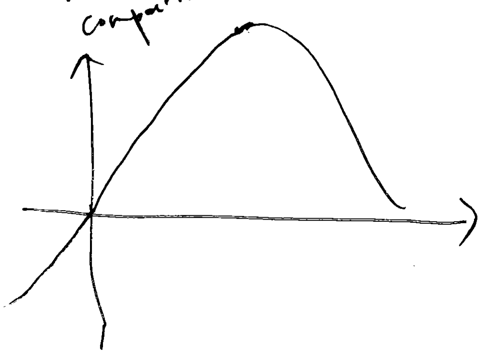




IGM($\phi_A - \phi_B$)



Microwave phase comparison



Conceptual steps:

Recall "~~p~~ ~~spec~~ phase comparator" view of dual-comb LOS:

Very precise knowledge of pulse temporal overlap when interference

BUT

Have no information if they don't overlap when no interference

↳ requires scanning of the relative delay phase to get updates.

How do we ~~prov~~ use/p this to do ~~[illegible]~~ synchronization then?

→ They are momentarily synced ~~at~~ then walk off all the time

→ Could claim that we know their time offset at all time to within 100fs.,

BUT

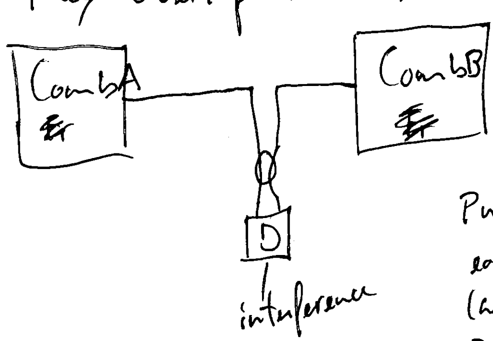
→ Much easier if we ~~had~~ actually had two combs ticking at the same rate, one at each side. Then we can prove that they tick at the same time within 100fs ~~by~~ because we can make them interfere continuously.

↳ Needs 3 combs total: f_r at site A

f_r at site B

$f_r + \Delta f_r$ at site B

How do we plan on demonstrating: ~~Have~~ Prove that the two combs tick at the same time ~~at~~ because they overlap in time at some calibration plane.



Put that in earlier slide. Cartoon with pulses being emitted.

What does sync mean in this case?
Why do we need three combs?
how will we prove time sync to pfs

Concepts: 1. LOS receiver measures pulse overlap or not
no interference if delay >100fs
interference if delay <100fs.

Key to precision + accuracy of pulse overlap

2. To measure other delays, need to scan delay axis:
Dual-comb ~~detuning~~: F_r and $F_r + \Delta F$
200MHz 200MHz + ~1-3kHz.

3. Measurement system is incompatible with synchronized combs: introduce third comb. Comb A and Comb C will be synchronized remotely. Comb B is used for scanning.

4. Will prove time sync through LOS.

"Phase comparison" formalism

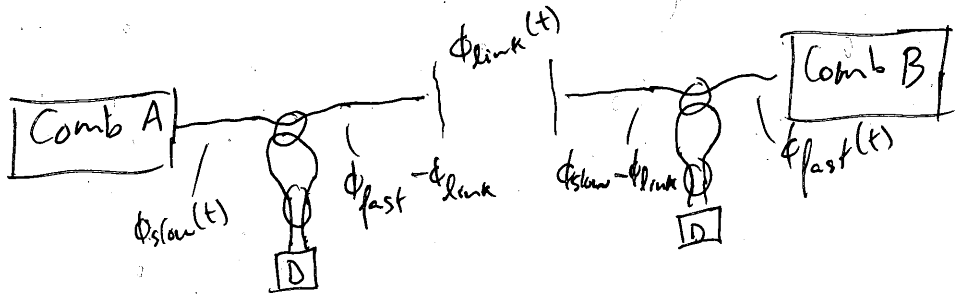
5. From IGMs, we get the phase difference between the two pulse trains. "From $I(t)$ to $\phi_{IGM}(t)$".

6. Equations are very simple with phases.
C.S Revisit one-way and two-way comparison with phases formalism?

Gives the equations

7. Details... of three-way phase comparison.

8. Difficulties:
- Fairly complex real-time system
  - IGM to phase processing
  - Communication link
- Needs "coarse" time sync first to be able to subtract the phases ~1ns accuracy
- Transceivers Path difference calibration (really just an overall phase offset): Fairly straightforward.



Site A:

$$\phi_{igm}(t) = (\phi_{fast} - \phi_{link}) - \phi_{slow}$$

$$\phi_{IGMA}(t) = \Delta\phi(t) - \phi_{link}(t)$$

Site B:

$$\phi_{igm}(t) = \phi_{fast} - (\phi_{slow} - \phi_{link})$$

$$\phi_{IGMB}(t) = \Delta\phi(t) + \phi_{link}(t)$$

$$\phi_{IGMA}(t) + \phi_{IGMB}(t) = 2\Delta\phi(t)$$: insensitive to ϕ_{link}

But assumes we can line up $\phi_{IGMA}(t)$ and $\phi_{IGMB}(t)$ to subtract

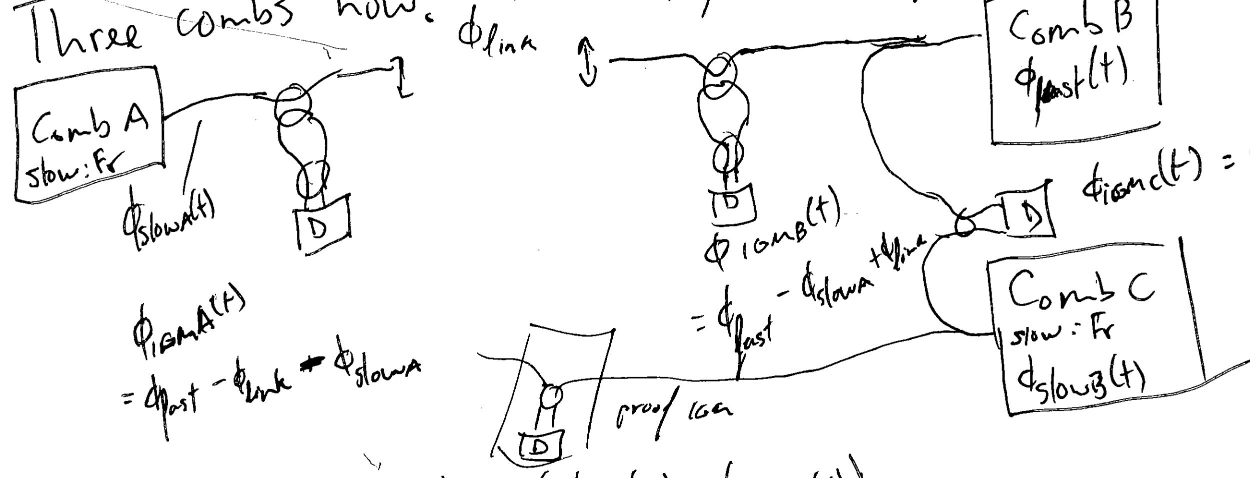
Need to have a common timebase for this comparison.

How good does it need to be?

Slope of $\phi_{IGMA}(t)$ controls sensitivity.

Three-way phase comparison.

Three combs now:



$$\phi_{IGMA}(t) + \phi_{IGMB}(t) = 2(\phi_{fast}(t) - \phi_{slowA}(t))$$

$$\phi_{IGMC}(t) = \phi_{fast}(t) - \phi_{slowB}(t)$$

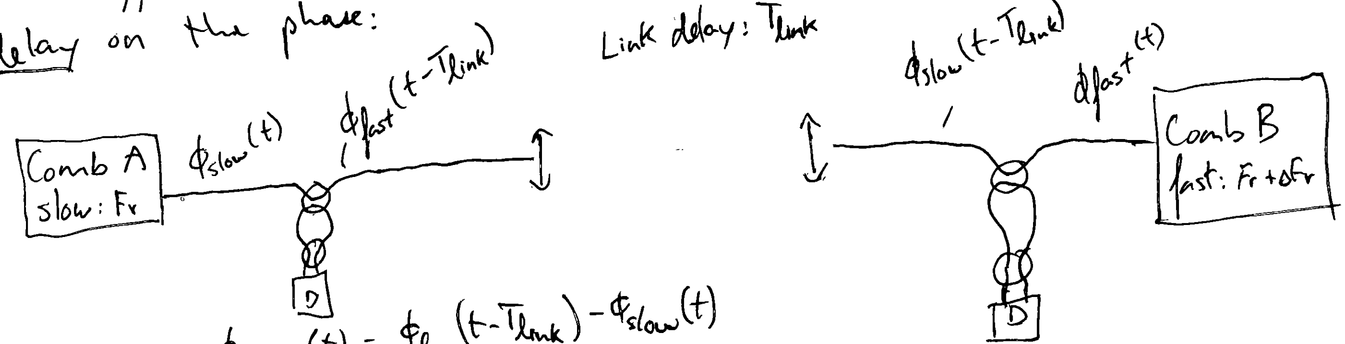
$$\cancel{2\phi_{IGMC}(t)}\ \phi_{IGMA}(t) + \phi_{IGMB}(t) - 2\phi_{IGMC}(t) = 2\phi_{slowB}(t) - 2\phi_{slowA}(t)$$

BUT:
- Ambiguity time cut in half b/c we compared 2ϕ — 2.5ns instead of 5ns
- Path offsets in the transceivers

Assumes no timestamp error

The effect of the link is not a <u>phase offset</u>: it's really a <u>delay</u> on the phase:

Link delay: T_{link}



$$\phi_{IGMA}(t) = \phi_{fast}(t-T_{link}) - \phi_{slow}(t)$$

$$\phi_{IGMB}(t) = \phi_{fast}(t) - \phi_{slow}(t-T_{link})$$

Assume $\phi_{slow}(t) = 2\pi f_r t + \phi_{oslow}$

$$\phi_{fast}(t) = 2\pi(f_r + \Delta f_r)t + \phi_{ofast}$$

Then: $\phi_{slow}(t-T_{link}) = 2\pi f_r t + \phi_{oslow} - 2\pi f_r T_{link} = \phi_{slow}(t) - 2\pi f_r T_{link}$

$$\phi_{fast}(t-T_{link}) = 2\pi(f_r+\Delta f_r)t + \phi_{ofast} - 2\pi(f_r+\Delta f_r)T_{link} = \phi_{fast}(t) - 2\pi(f_r+\Delta f_r)T_{link}$$

$$\phi_{IGMA}(t) = \phi_{fast}(t) - 2\pi(f_r+\Delta f_r)T_{link} - \phi_{slow}(t)$$

$$\phi_{IGMB}(t) = \phi_{fast}(t) - \phi_{slow}(t) + 2\pi f_r T_{link}$$

$$\phi_{IGMA}(t) + \phi_{IGMB}(t) = 2\left(\phi_{fast}(t) - \phi_{slow}(t)\right) - 2\pi\Delta f_r T_{link}$$

~10e3 less sensitive to one-way fluctuations than one IGM but still...

1. Can we scale $\phi_{IGMA}(t)$ or $\phi_{IGMB}(t)$ before summing them in a meaningful manner even though all the equations are modulo 2π?

2. Can we time shift $\phi_{IGMA}(t)$ and/or $\phi_{IGMB}(t)$ before summing them?

$$\phi_{IGMA}(t-T_{shiftA}) = \phi_{fast}(t-T_{shiftA}) - \phi_{slow}(t-T_{shiftA}) - 2\pi(f_r+\Delta f_r)T_{link}$$

$$= \phi_{fast}(t) - \phi_{slow}(t) - 2\pi(f_r+\Delta f_r)T_{shiftA} + 2\pi f_r T_{shiftA} - 2\pi(f_r+\Delta f_r)T_{link}$$

$$= \phi_{fast}(t) - \phi_{slow}(t) - 2\pi\Delta f_r T_{shiftA} - 2\pi(f_r+\Delta f_r)T_{link}$$

$$\phi_{IGMB}(t-T_{shiftB}) = \phi_{fast}(t-T_{shiftB}) - \phi_{slow}(t-T_{shiftB}) + 2\pi f_r T_{link}$$

$$= \phi_{fast}(t) - 2\pi(f_r+\Delta f_r)T_{shiftB} - \phi_{slow}(t) + 2\pi f_r T_{shiftB} + 2\pi f_r T_{link}$$

$$= \phi_{fast}(t) - \phi_{slow}(t) - 2\pi\Delta f_r T_{shiftB} + 2\pi f_r T_{link}$$

$\phi_{1GMA}(t - T_{shiftA})$ ~~$\phi_{fast}(t)$~~ $+ \phi_{1GMB}(t - T_{shiftB}) =$

$$2\left(\phi_{fast}(t) - \phi_{slow}(t)\right) - 2\pi \Delta f_r \left(T_{shiftA} + T_{shiftB}\right) - 2\pi \Delta f_r T_{link}$$

So if $T_{shiftA} + T_{shiftB} = -T_{link}$, this effect cancels out. Is this even possible in real time?

3. Other possibility: If the link stability is enough and all we care about is having something which is insensitive to link distance even for the phase offset, we can use the coarse T_{link} and do:

$\phi_{1GM_A}(t) + \phi_{1GM_B}(t) + 2\pi \Delta f_r T_{link,coarse}$

In time units, this is a shift of $\frac{2\pi \Delta f_r}{2\pi f_r} T_{link}$

$T_{link} = \frac{2\,km}{3e8\,m/s} \approx \frac{3km}{3e8\,m/s} = \frac{3e3}{3e8} = 1e-5\,s = 10\,\mu s$ 100ps!

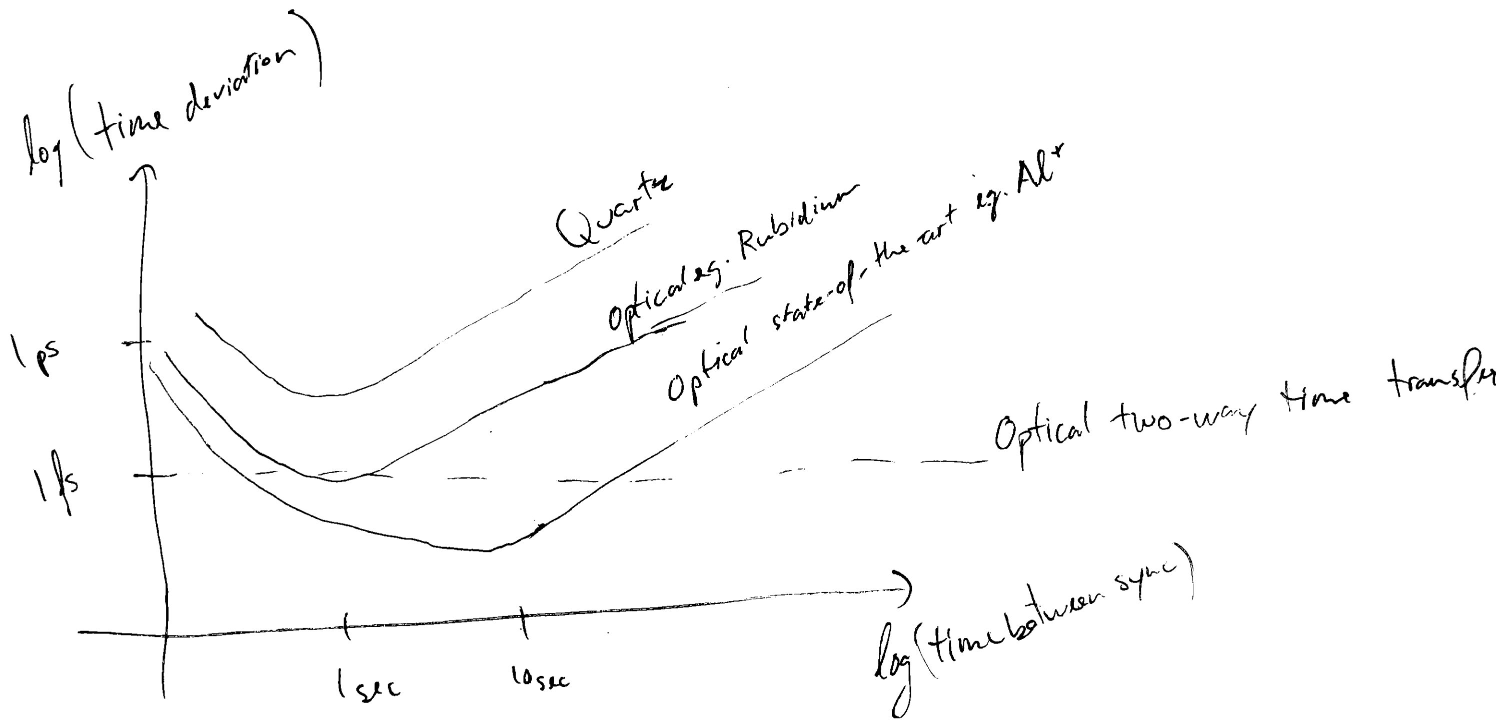
$T_{link} \cdot \frac{\Delta f_r}{f_r} = 10\,\mu s \cdot \frac{2\,kHz}{200\,MHz} = 10\,\mu s \cdot \frac{2e3}{2e8} = \frac{10\,\mu s}{1e5} = \frac{1e-5\,s}{1e5} = 1e-10\,s = 100\,ps$ ... this is huge

Very noticeable...

What about the other way of seeing it: "Both 1GMs move equal amounts in opposite direction." Is this strictly true ~~or~~ or is there a gain difference of 1 vs $1 + \frac{\Delta f_r}{f_r}$?

How often do we need to re-sync to the master clock?

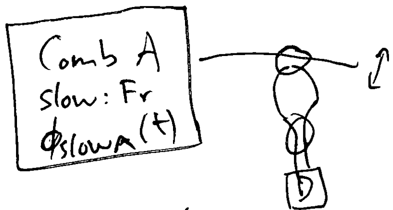
Depends on clock technology and desired sync performance.



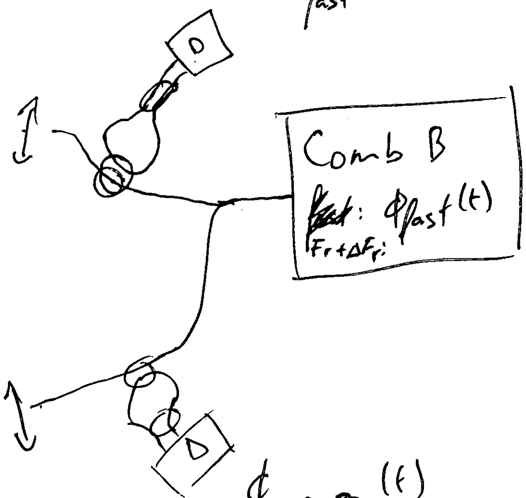
Three combs, relay in the middle:

$\phi_{IGMB1}(t) = \phi_{fast}(t) - \left(\phi_{slowA}(t) - \phi_{link1}\right)$

$= \phi_{fast}(t) - \phi_{slowA}(t) + \phi_{link1}$



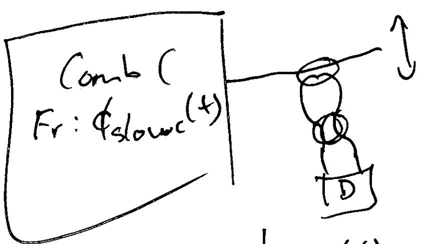
ϕ_{link1}



$\phi_{IGM_A}(t)$

$= \phi_{fast}(t) - \phi_{link1} - \phi_{slowA}(t)$

ϕ_{link2}



$\phi_{IGMB2}(t)$

$= \phi_{fast} - \left(\phi_{slowC} - \phi_{link2}\right)$

$\phi_{IGMC}(t)$

$= \phi_{fast}(t) - \phi_{link2} - \phi_{slowC}$

$\phi_{IGMA} - \phi_{IGMC} = \phi_{slowC} - \phi_{slowA} + \phi_{link2} - \phi_{link1}$

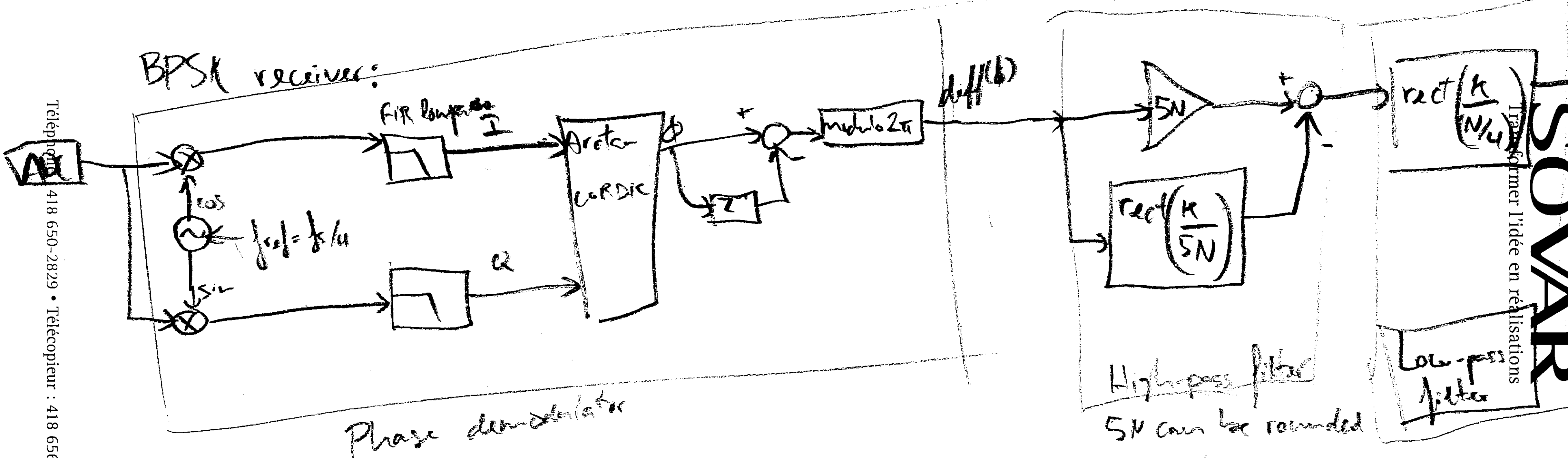
Independent of ϕ_{fast} but has $\phi_{link2} - \phi_{link1}$

→ If Comb A and C are at the same site (which they are otherwise there is no ground truth), we need only one comm link.

$\phi_{IGMB1} - \phi_{IGMB2} = \phi_{slowC} - \phi_{slowA} + \phi_{link1} - \phi_{link2}$

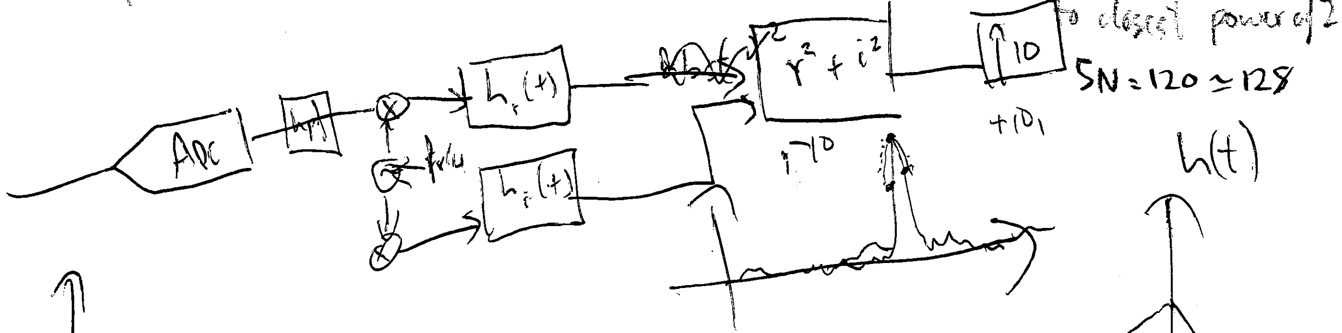
$\left(\phi_{IGMA} - \phi_{IGMC}\right) + \left(\phi_{IGMB1} - \phi_{IGMB2}\right) = 2\left(\phi_{slowC} - \phi_{slowA}\right)$

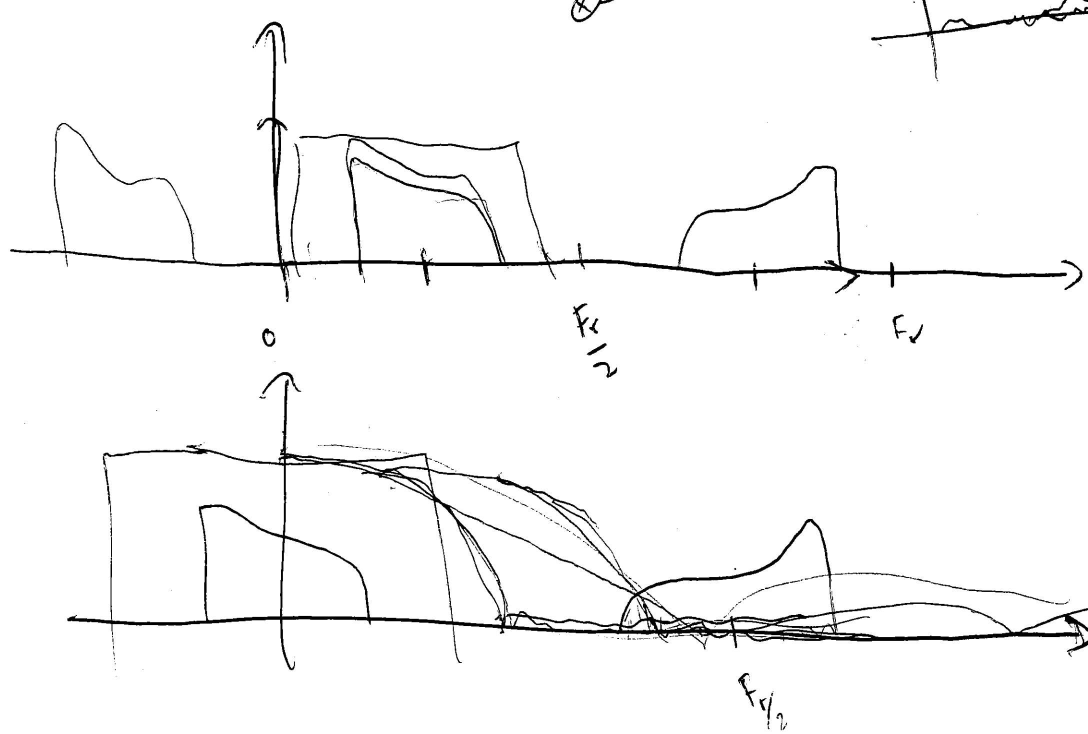
BPSK receiver:

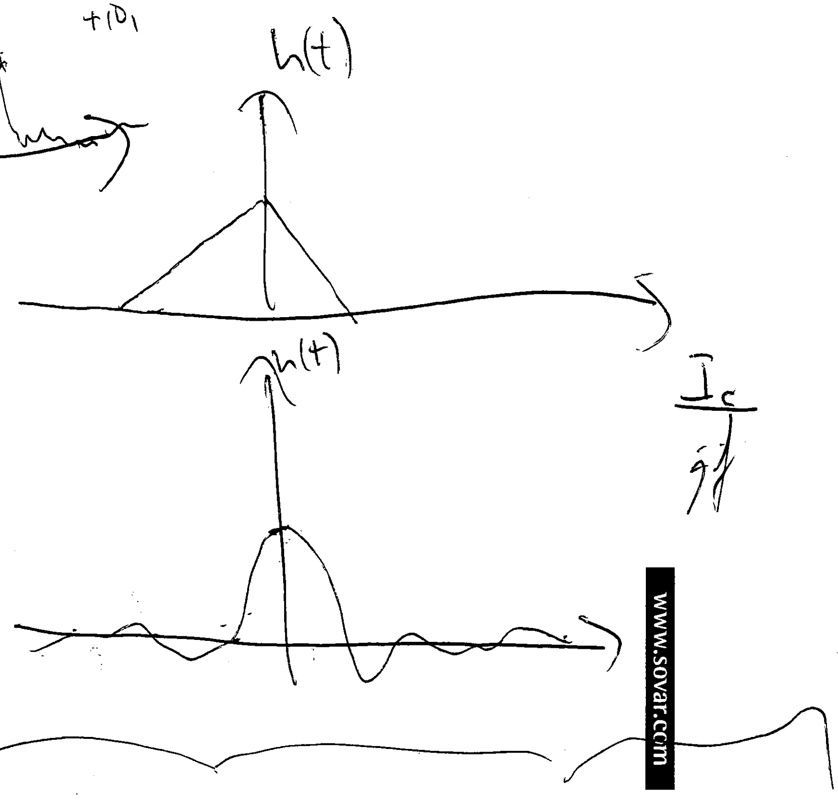


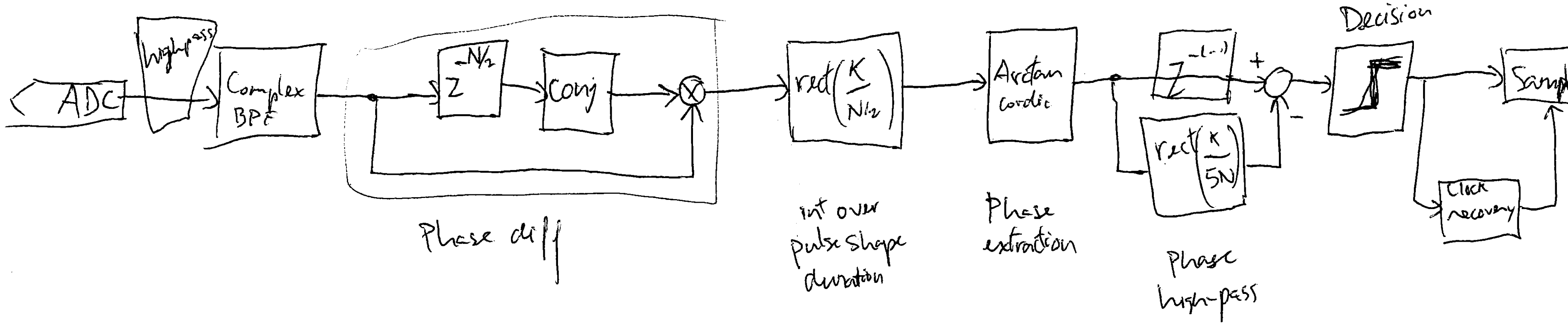
5N can be rounded to closest power of 2

5N = 120 ≃ 128

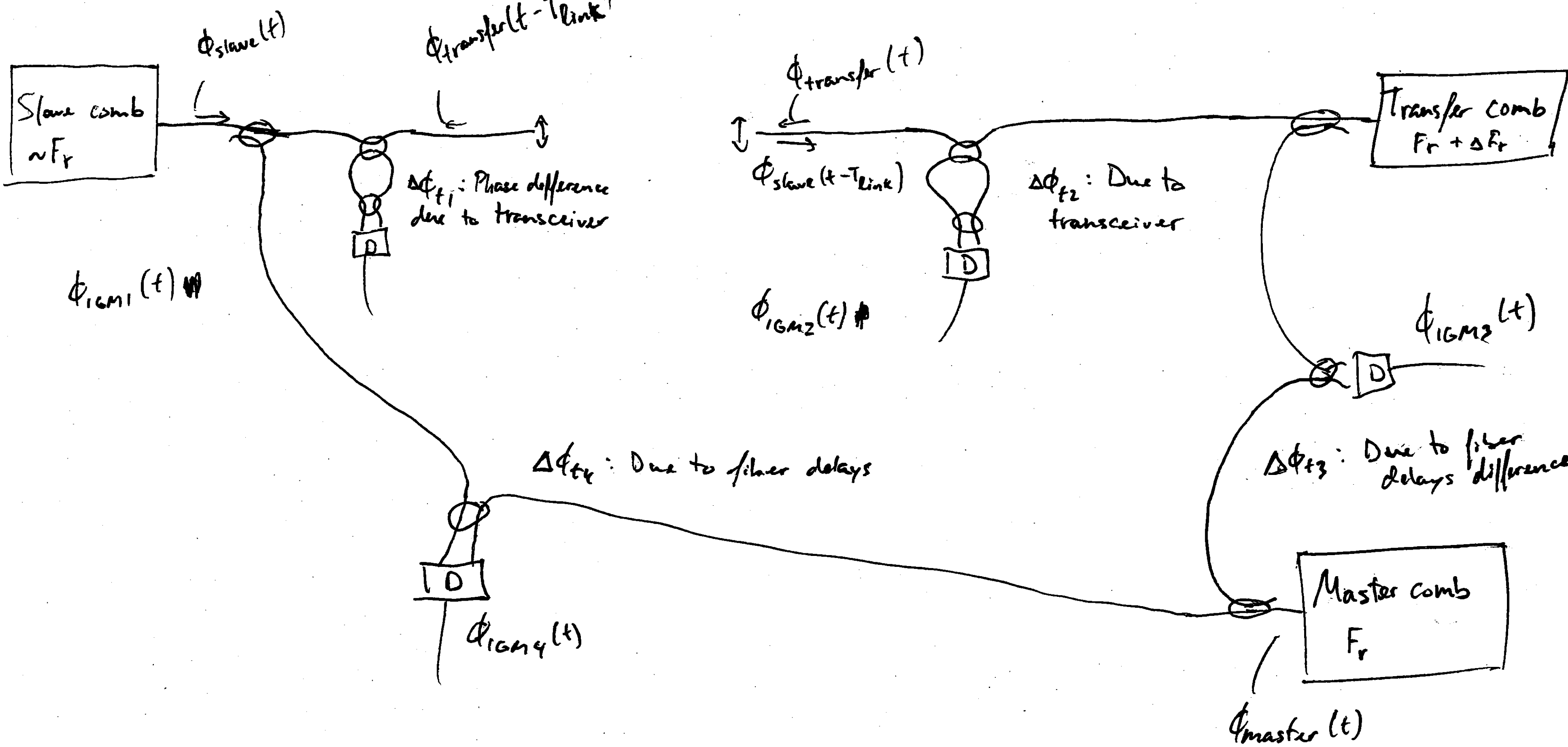








ADC
highpass
Complex BPF
$Z^{-N/2}$
Conj
Phase diff
$\mathrm{rect}\left(\frac{K}{N/2}\right)$
int over pulse shape duration
Arctan cordic
Phase extraction
$Z^{-(\ldots)}$
$\mathrm{rect}\left(\frac{K}{5N}\right)$
Phase high-pass
Decision
Clock recovery



Derivation of Synchronization equations

$$\phi_{IGM1}(t) = \phi_{transfer}(t - T_{link}) - \phi_{slave}(t) + \Delta\phi_{t1}$$

$$\phi_{IGM2}(t) = \phi_{transfer}(t) - \phi_{slave}(t - T_{link}) + \Delta\phi_{t2}$$

$$\phi_{IGM3}(t) = \phi_{transfer}(t) - \phi_{master}(t) + \Delta\phi_{t3}$$

Measurement equations

$$\phi_{IGM4}(t) = \phi_{slave}(t) - \phi_{master}(t) + \Delta\phi_{t4}$$

Only accessible at calibration step, afterwards becomes validation.

Generate the linear combination:

$\tilde{T}_{link}$ is the link distance estimate obtained by the coarse TWTFT.

$$\phi_m(t) \equiv \phi_{IGM1}(t) + \phi_{IGM2}(t) - 2\phi_{IGM3}(t) + 2\pi\Delta f_r \tilde{T}_{link}$$

① $$\phi_m(t) = \phi_{transfer}(t) + \phi_{transfer}(t - T_{link}) - \phi_{slave}(t) - \phi_{slave}(t - T_{link}) + \Delta\phi_{t1} + \Delta\phi_{t2} - 2\phi_{transfer}(t) - 2\phi_{master}(t) - \Delta\phi_{t3}$$

Assume: $\phi_{slave}(t) = \phi_{o,slave} + 2\pi f_r t$

② $$\phi_{slave}(t - T_{link}) = \phi_{slave}(t) - 2\pi f_r T_{link}$$

likewise: $\phi_{transfer}(t) = \phi_{o,transfer} + 2\pi(f_r + \Delta f_r)t$

③ $$\phi_{transfer}(t - T_{link}) = \phi_{transfer}(t) - 2\pi(f_r + \Delta f_r)T_{link}$$

Put ② and ③ into ①:

$$\phi_m(t) = 2\phi_{transfer}(t) - 2\phi_{slave}(t) - 2\pi\Delta f_r T_{link} - 2\phi_{transfer}(t) + 2\phi_{master}(t) + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} + 2\pi\Delta f_r \tilde{T}_{link}$$

$$\phi_m(t) = 2\phi_{master}(t) - 2\phi_{slave}(t) \underbrace{- 2\pi\Delta f_r T_{link}}_{\text{Measured by coarse TWTFT}} + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} + 2\pi\Delta f_r \tilde{T}_{link}$$

$$\phi_m(t) = 2(\phi_{master}(t) - \phi_{slave}(t)) + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} - 2\pi\Delta f_r(T_{link} - \tilde{T}_{link})$$

At calibration step we force:

$$\phi_{10my}(t) = 0 \Rightarrow \phi_{master}(t) - \phi_{slave}(t) = \Delta\phi_{t4} \quad \Big\}$$ This is detected by interference between the master and the slave. at the reference detector.

This gives for $\phi_m(t)$:

$$\phi_m = 2\Delta\phi_{t4} + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} - 2\pi\Delta f_r\left(T_{link} - \tilde{T}_{link}\right) \equiv \phi_{m,ref}$$

This becomes our reference phase to which we will lock.

Assuming that none of the transceivers and out-of-loop fibers drift, we have at measurement time:

$$\phi_{m2}(t) = 2\left(\phi_{master}(t) - \phi_{slave}(t)\right) + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} - 2\pi\Delta f_r\left(T_{link2} - \tilde{T}_{link2}\right)$$

By locking this signal to $\phi_{m,ref}$, we get:

$$\phi_{m2}(t) = \phi_{m,ref}$$

$$2\left(\phi_{master}(t) - \phi_{slave}(t)\right) + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} - 2\pi\Delta f_r\left(T_{link2} - \tilde{T}_{link2}\right) = 2\Delta\phi_{t4} + \Delta\phi_{t1} + \Delta\phi_{t2} - \Delta\phi_{t3} - 2\pi\Delta f_r\left(T_{link} - \tilde{T}_{link}\right)$$

$$2\left(\phi_{master}(t) - \phi_{slave}(t)\right) - 2\Delta\phi_{t4} = 2\pi\Delta f_r \underbrace{\left(T_{link2} - \tilde{T}_{link2} - T_{link} + \tilde{T}_{link}\right)}$$

Hopefully 0. Note that $\tilde{T}_{link}$ can even have a bias, as long as it is the same in the calibration step as the real measurement, it drops out.

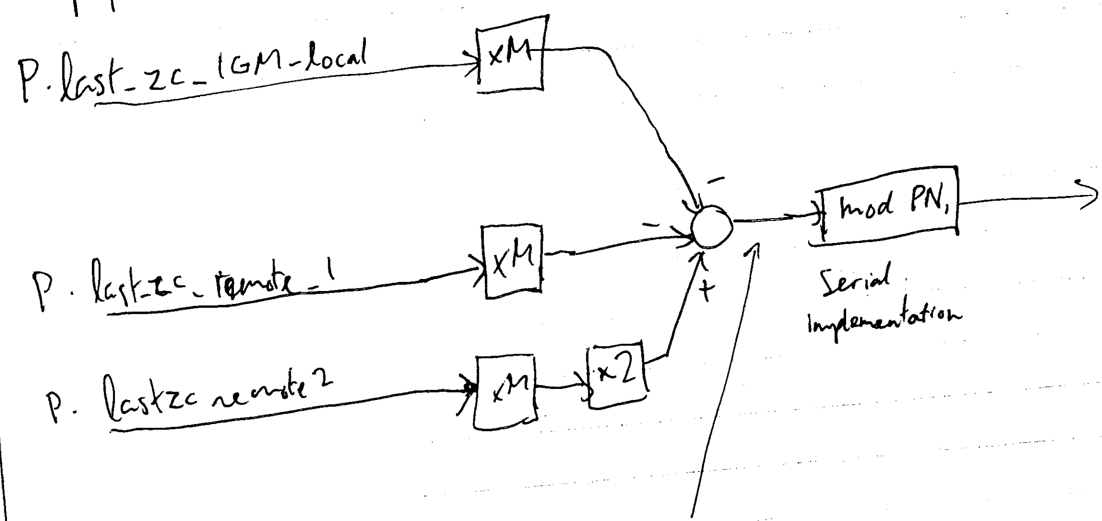
$$2\left(\phi_{master}(t) - \phi_{slave}(t)\right) - 2\Delta\phi_{t4} = 0$$

Which also implies:

$$2\,\phi_{10my}(t) = 2\left(\phi_{slave}(t) - \phi_{master}(t) + \Delta\phi_{t4}\right) = 0,$$

which means interference between the master's and the slave's pulses, ~~[illegible]~~ up to a π phase offset (because we are locking $2(\phi_{master}(t) - \phi_{slave}(t))$). This ambiguity is resolved by the coarse TWTFT, Implementation of synchronization equations

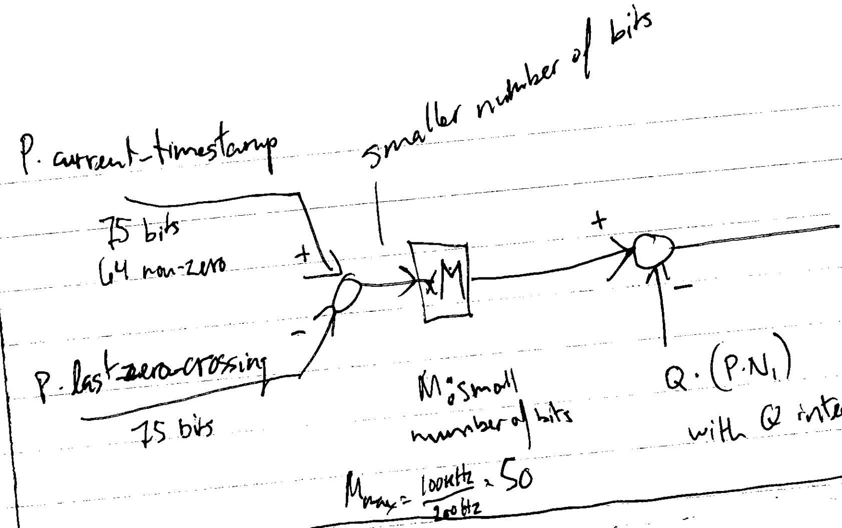
Simplification:



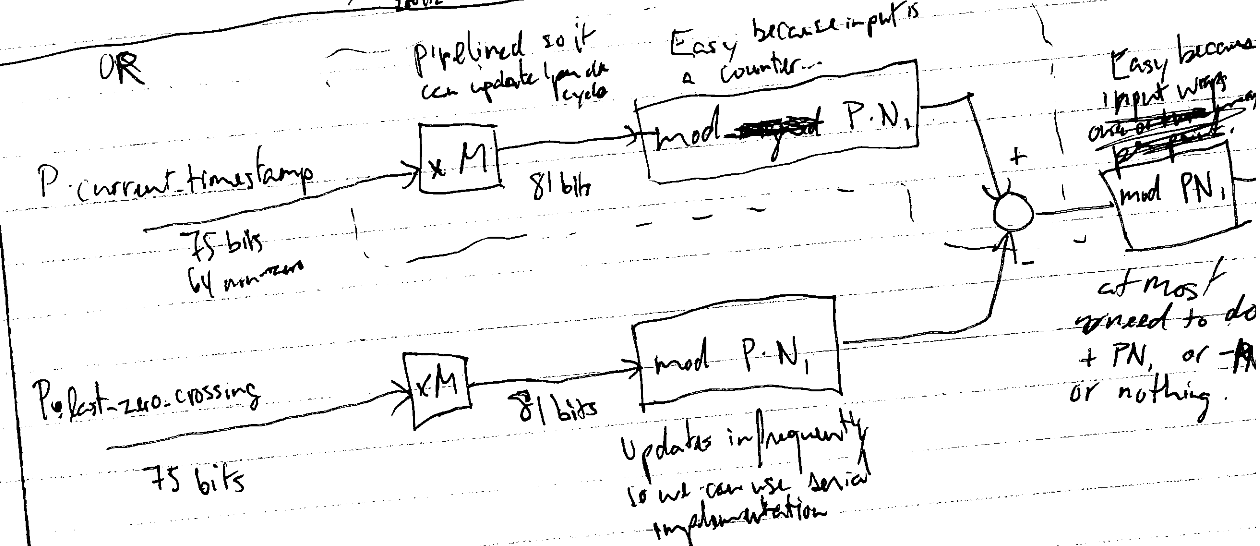
This should already be a small number so the serial modulo shouldn't be a problem.

For syntonization, we have to do it differently. Two reasons: 1. We won't have the last branch which handles the subtraction of any large offset.
2. The modulus is different for 1GM-local and 1GM_remote since the clock rates are different.

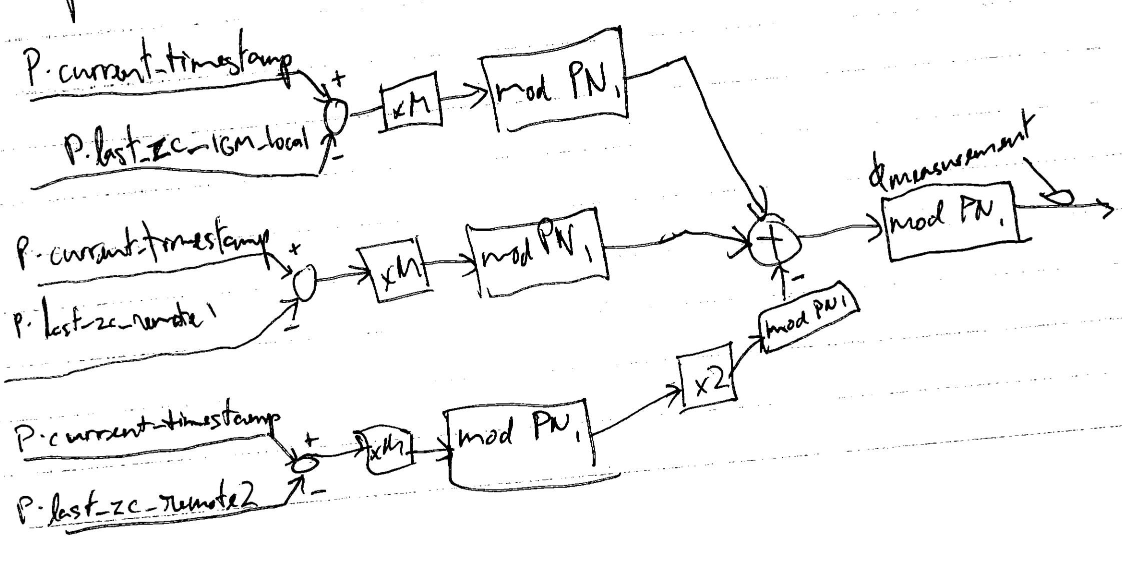
$$\frac{1e5}{2e2} = \frac{1}{2}e2 = 50 \qquad \log_2(50) < \log_2 64 = 6 \text{ bits}$$



OR



I just realized that it might be useless to compute the P. current_timestamp branch since they will all cancel once we combine the phases from other IGMs... It would be so nice.



What we want is the current phase, modulo 2π.
We really carry only the digital phase, $\phi_{digital} = \frac{\phi_{analog}}{2\pi} \cdot Radix$, where $Radix = N_1 \cdot P$, $P = 2^{10}$

$$[counts] = [rad]\left[\frac{counts}{rad}\right]$$

$\phi_{digital}$ is modulo $N_1 \cdot P$.
Instead we measure the timestamps, which are modulo $K = \frac{N_1}{M}$, but they really are $P \cdot$ timestamps since we want to hold the fractional part too, making them mod $P \cdot K = \frac{P \cdot N_1}{M}$
So to get the current phase of one IGM we do:

$$(P \cdot current\_timestamp - P \cdot last\_zero\_crossing) \mod P \cdot N_1 / M$$

This would be scaled in $\frac{\phi_{analog}}{2\pi} \cdot \frac{P \cdot N_1}{M}$ and would be modulo $\frac{P \cdot N_1}{M}$.

If instead we did:

$$(PM \cdot current\_timestamp - PM \cdot last\_zero\_crossing) \mod P \cdot N_1$$

At least the modulo would be tractable. We could simplify this to:

$$(M(P(current\_timestamp - last\_zero\_crossing))) \mod P N_1$$

and then that means we could compute the subtraction at the $P \cdot$ timestamps bitwidth (75 bits), then multiply the timestamp (which we could now restrict a little bit without wrapping) by the small number M. Then we could apply the modulo operation on a reasonable number of bits.

its even better than this since current_timestamp would have P zeros so it would really be a 64 bits subtraction.

This gives us a phase for one of the IGM scaled as

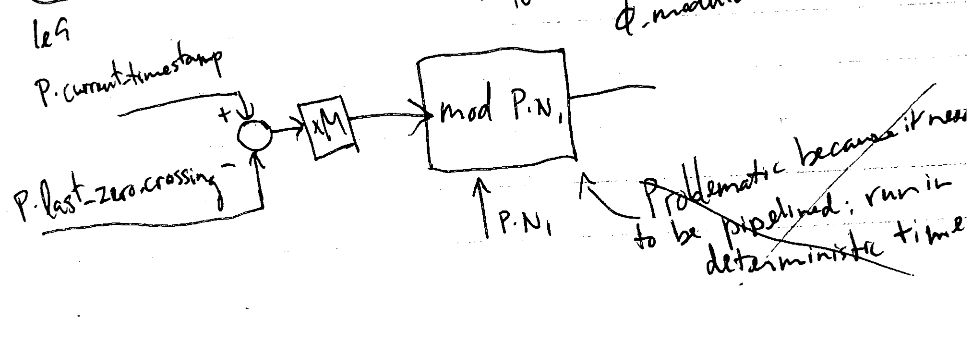
$$\phi_{digital} = \frac{\phi_{analog}}{2\pi} \cdot \underbrace{P \cdot N_1}_{} = \frac{\phi_{analog}}{2\pi} \cdot P \cdot M \cdot K \mod (P N_1)$$

Effectively $\simeq 2^{10} \cdot 1e6 \simeq 2^{30} \simeq 1e9$

So that our resolution is

$$\frac{5ns}{1e9} \simeq 5as.$$

Needs to be rewritten to really implement mod_signed().



What we want at a high level:

$$\phi_m \equiv \phi_{IGM1} + \phi_{IGM2} - 2\phi_{IGM3} + 2\pi\Delta f_r \tilde{T}_{link}$$

Measured as a local timestamp

Remote timestamps, need to adjust by ΔT_{AB}

Measured as

$$N_{link} = F_r \tilde{T}_{link} \cdot P \cdot 2$$

$$= \Delta f_r \tilde{T}_{link} \left(\frac{F_r}{\Delta f_r}\right) \cdot P \cdot 2$$

factor of 2 b/c of two-way

$P = 2^{10}$ scaling to add sub-sample resolution to our timestamps

$K_1 = \frac{F_r}{\Delta f_r}$ Number of points per IGM

$N_1 = M \cdot K_1$ Mode number of the master comb's CW lock.

$M = N_2 - N_1$ Mode number difference of the CW locks between "transfer" and "master" combs.

$$\phi_{IGM1} = \frac{1}{2\pi}\phi_{analog} \cdot P \cdot K_1 \mod PK_1$$

~~or~~

$$\cancel{\phi_{IGM1} = \phi_{analog}}$$

$\frac{F_r}{\Delta f_r} = K_1$ Number of points per IGM.

$$N_{link} = K_1 \Delta f_r \tilde{T}_{link} \cdot P \cdot 2$$

$$\phi_{link,d} = \frac{M}{2} \cdot N_{link} = \Delta f_r \tilde{T}_{link} \cdot PK_1M$$

Same scaling.

But we will multiply by M to make the modulus an integer:

$$\phi_{IGM1} = \frac{1}{2\pi}\phi_{analog} \cdot P \cdot K_1 \cdot M \mod PK_1M$$

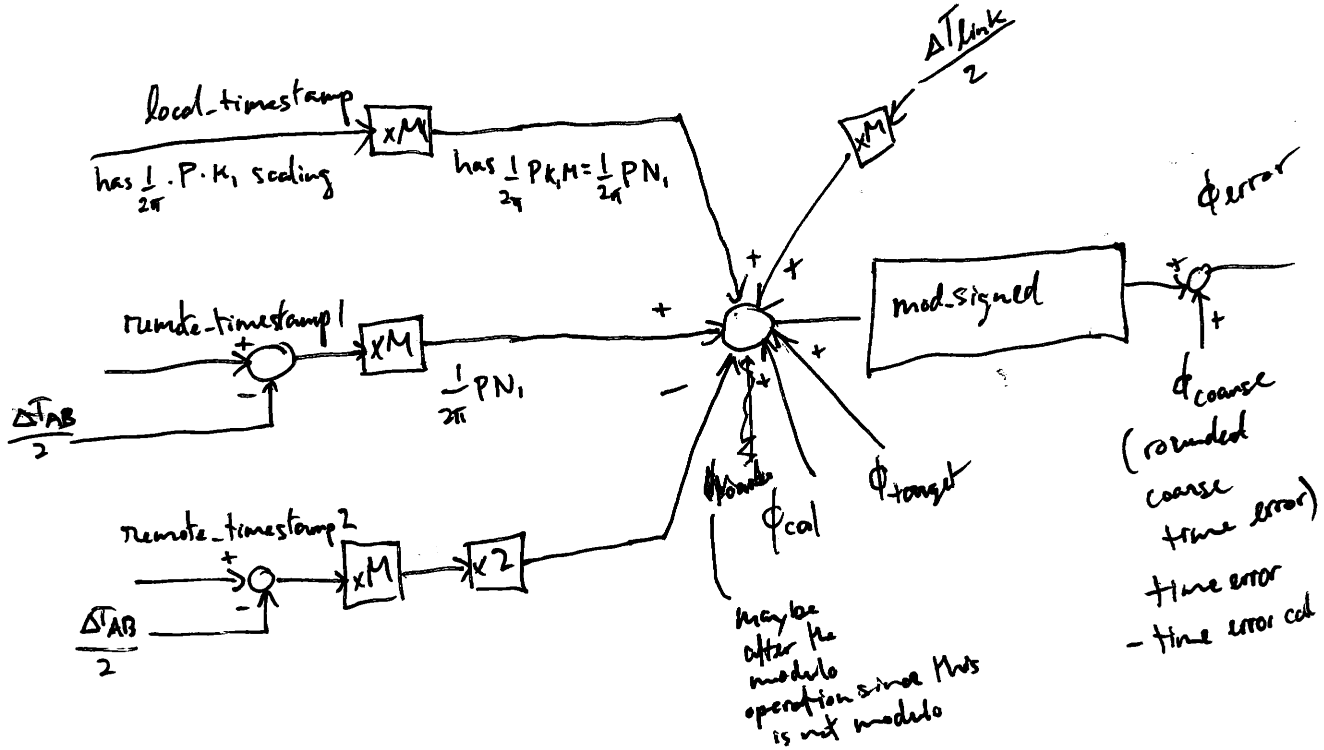
$$= \frac{1}{2\pi}\phi_{analog} \cdot P \cdot N_1 \mod P \cdot N_1$$

ϕ_{IGM2} measured as a remote timestamp:

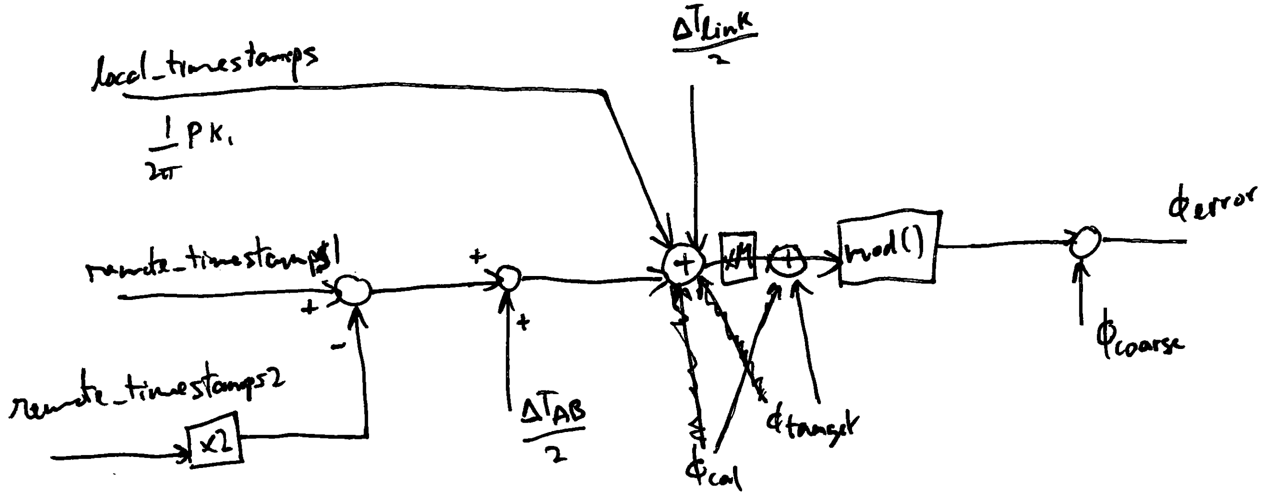
$$\phi_{IGM2,\,digital} = \frac{1}{2\pi}\phi_{analog} \cdot P \cdot K_1 \cdot M$$

So I need to do $\phi_{IGM2,\,digital} - \frac{\Delta T_{AB}}{2}$

ΔT_{AB} already has the same scaling, except for a factor of two.



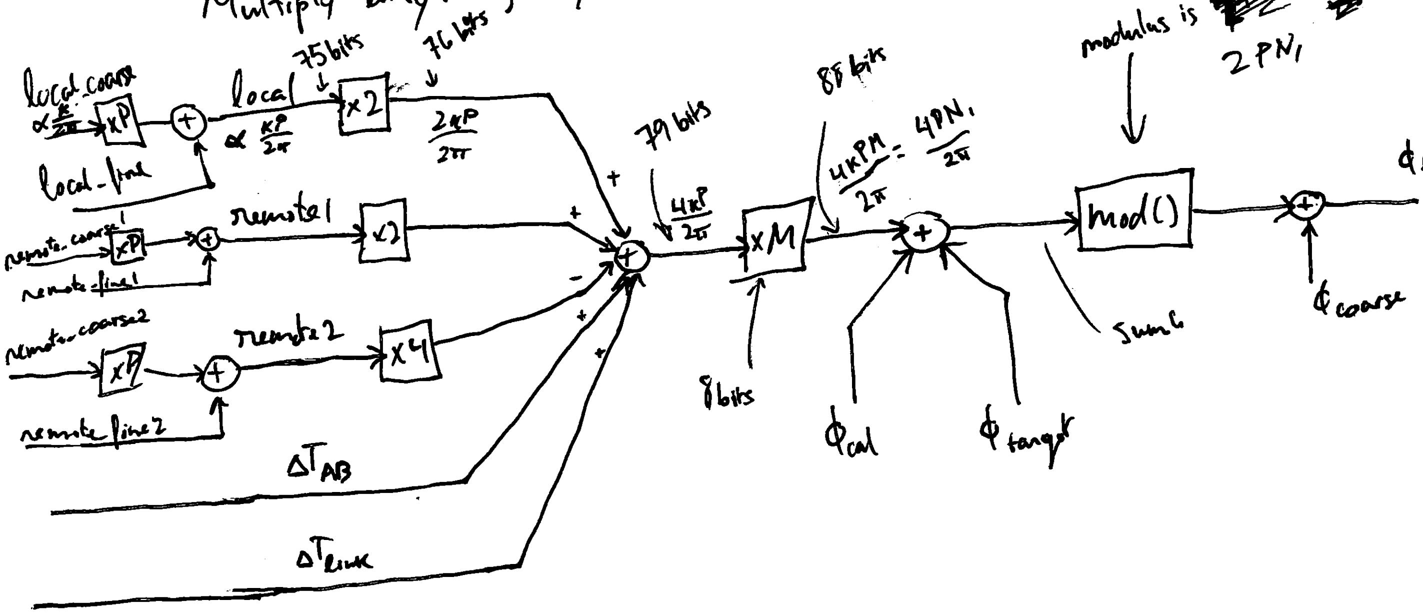
Push the xM ~~x~~ after summing:



Multiply everything by two to avoid having to divide by 2:

I think

$4 \cdot P \cdot N_1$ mod $2PN_1$



$\phi_{error} \propto \frac{\phi_{analog}}{2\pi} \cdot$

Note that the modulo is different by a factor of two than the scaling.

$$A \times B = \left(A_{high} \cdot 2^N + A_{low}\right) B = A_{high} B \cdot 2^N + A_{low} B$$

A: 80 bits signed; B: 8 bits unsigned; A_{high}: signed; A_{low}: unsigned; A_{high}: signed, B: unsigned; A_{low}: unsigned 40 bits, B: unsigned 8 bits

Choose $N = 40$

use 42 bits, signed × 8 bits unsigned multipliers.

What about the fact that we are signed?

$A \times B =$

$$\text{Pts per } 1\text{GM} = K = \frac{F_S}{\Delta F_r}$$

$F_S = F_{r\,purple}$

$\Delta F_r = F_{r\,orange} - F_{r\,purple}$

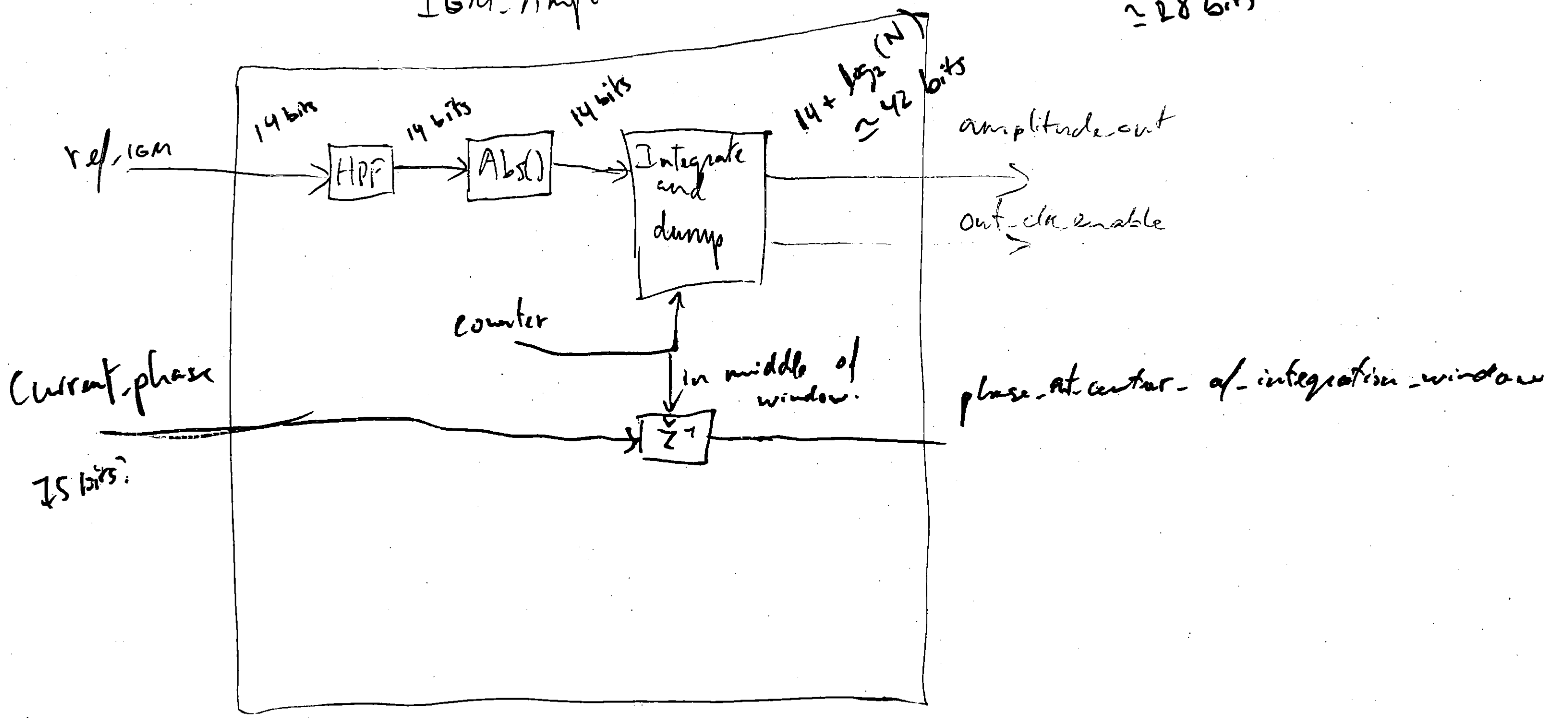
$$F_{r\,orange} = \frac{F_{cw}}{n_{orange}} \qquad F_{r\,purple} = \frac{F_{cw}}{n_{purple}}$$

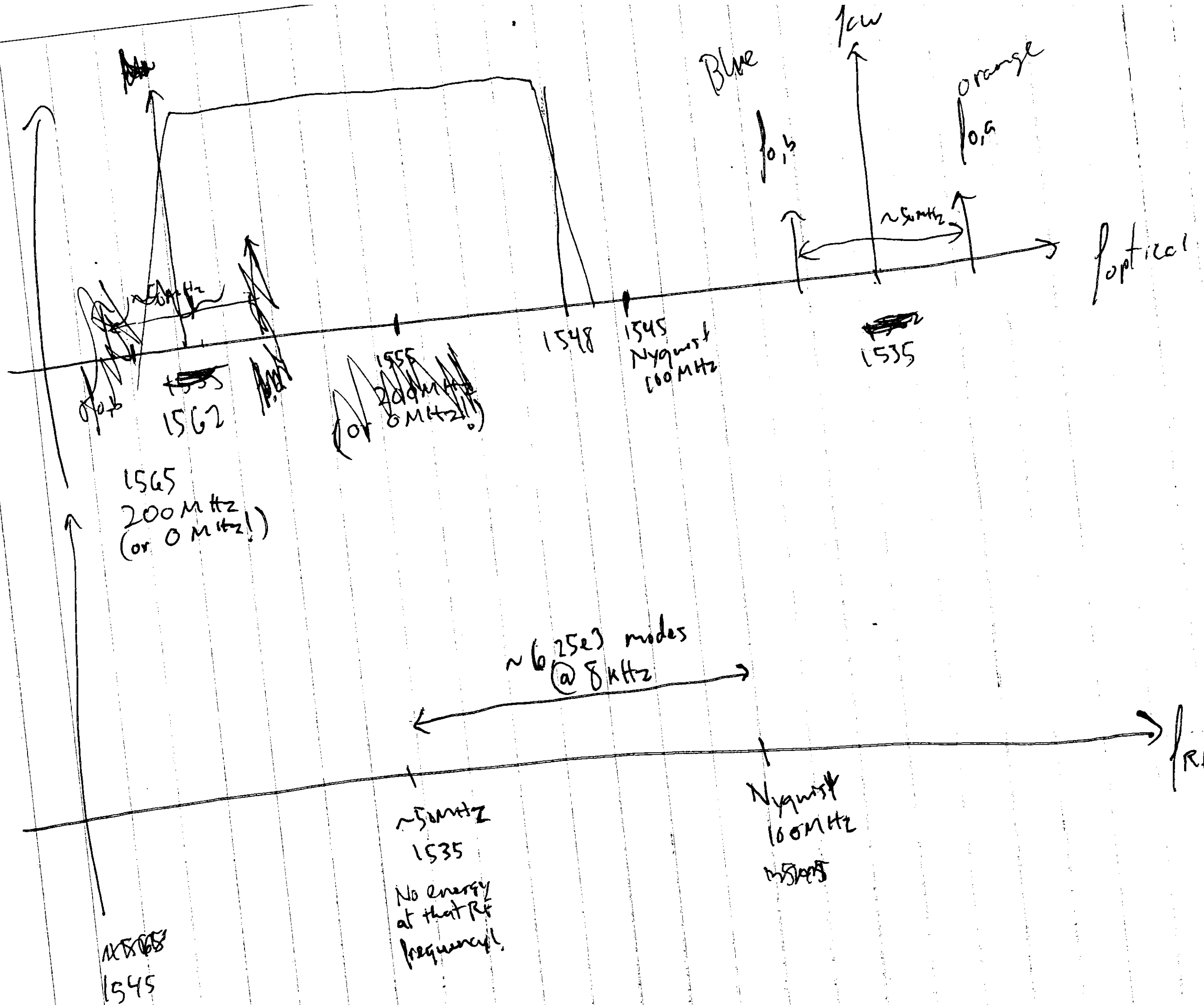
$$K = \frac{\cancel{F_{cw}}}{n_{purple}} \cdot \frac{1}{\left(\frac{\cancel{F_{cw}}}{n_{orange}} - \frac{F_{cw}}{n_{purple}}\right)} = \frac{1}{\frac{n_{purple}}{n_{orange}} - 1}$$

$$= \frac{n_{orange}}{n_{purple} - n_{orange}}$$

N ~ 1 sec max?
200e6
≃ 28 bits

IGM_Amplitude_detector




Blue
fcw
orange
f0,b
f0,a
~50MHz
foptical
1535
1545
Nyquist
100MHz
1548
1555
(or 0MHz!)
1562
1565
200MHz
(or 0MHz!)
~6.25e3 modes
@ 8kHz
fRF
~50MHz
1535
No energy
at that RF
frequency!
Nyquist
100MHz
1545

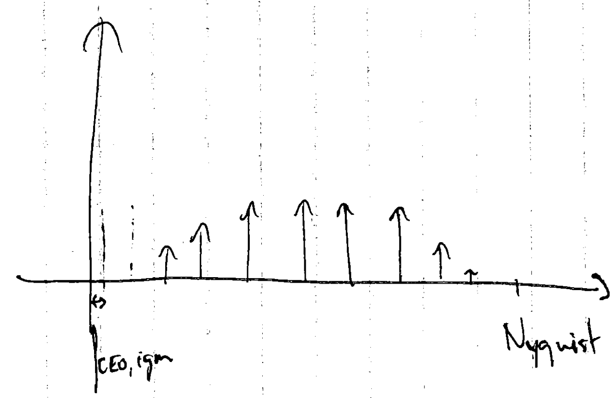
Frequency of every mode of both combs:

$$f_{CEO,orange} = -\frac{f_{r,orange}}{8}$$

$$f_{r,orange} = \frac{f_{cw}}{N_1}$$

$$f_{n,orange} = n f_{r,orange} + f_{CEO,orange}$$

$$\boxed{f_{n,orange} = \left(n - \tfrac{1}{8}\right) f_{r,orange}} \qquad f_{n,orange} = \left(\frac{n - \frac{1}{8}}{N_1}\right) f_{cw}$$



Blue: $f_{r,blue} = \frac{f_{cw}}{N_2}$

$$f_{m,blue} = m f_{r,blue} + f_{CEO,blue}$$

$$f_{CEO,blue} = +\frac{f_{r,blue}}{8}$$

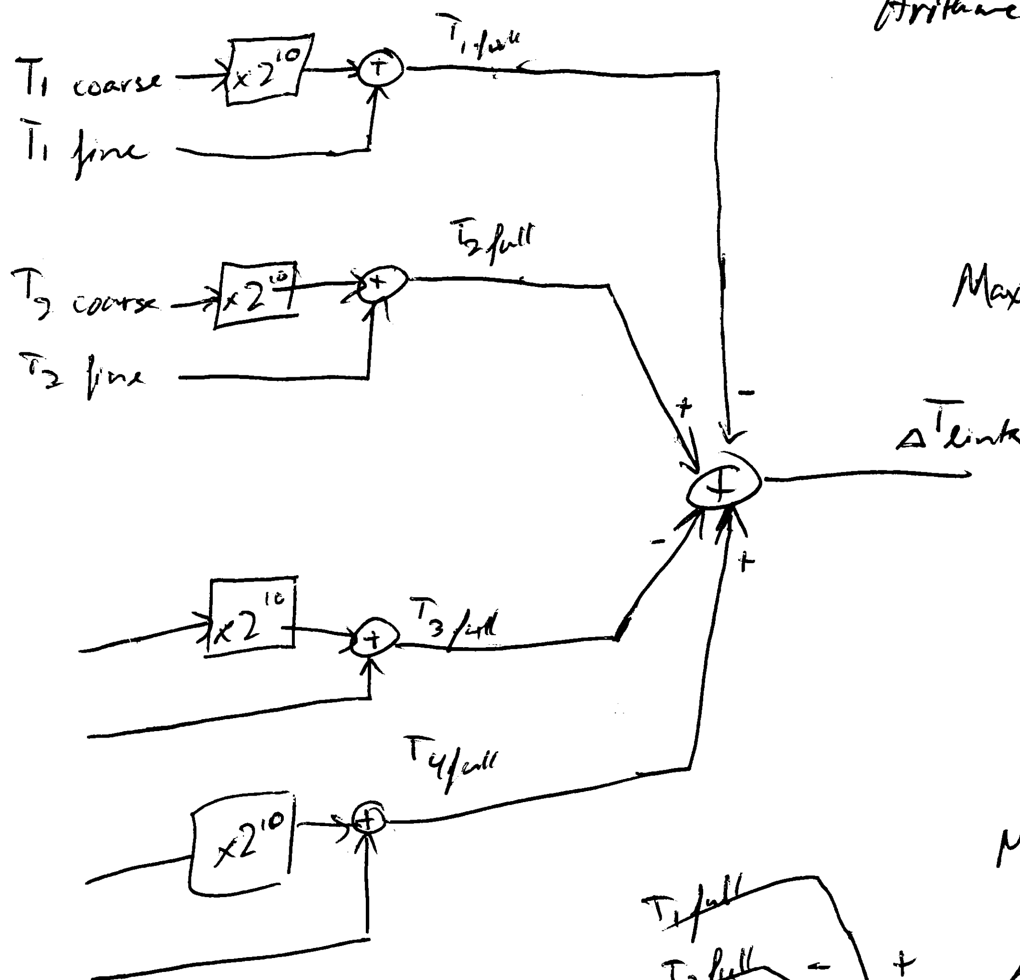
$$\boxed{f_{m,blue} = \left(m + \tfrac{1}{8}\right) f_{r,blue}} \qquad f_{m,blue} = \frac{\left(m + \frac{1}{8}\right)}{N_2} f_{cw}$$

RF beat frequencies are: $f(m,n) = \left|f_{n,orange} - f_{m,blue}\right| = f_{cw}\left|\frac{\left(n + \frac{1}{8}\right)}{N_1} - \frac{\left(m - \frac{1}{8}\right)}{N_2}\right|$ for all n, m (positive) integers

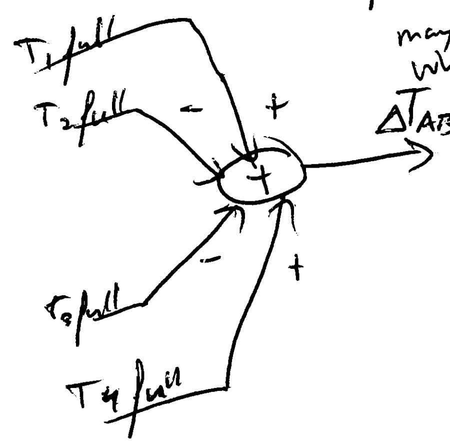
$N_1 =$

What are the closest frequencies to 0? There ~~[illegible]~~ is one for ~~the~~ each different Nyquist zone...

Arithmetic for coarse TWTFT

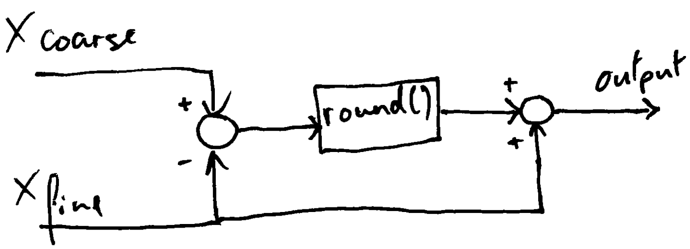


Max = Not too high...
1000Km / c = 3ms ≤ 1e6 samples
But keeping it fullscale gives a nice confirmation if things are working



Max = Very high...
maybe 1 day time offset at worst?
whatever let's keep it fullsize...

Combining a coarse, but non-modulo measurement with a more precise, but modulo measurement:



X_{coarse} and X_{fine} need to have been scaled to exactly the same units, with no offset between them.

X_{real} is the desired quantity

X_{fine} is the precise measurement, but modulo something

$$x_{fine} = x_{real} \text{ modulo } x_{modulus}$$
$$= x_{real} + \underset{\uparrow \text{ arbitrary integer}}{K} X_{modulus}$$

$$x_{coarse} = x_{real} + \underset{\uparrow}{x_{noise}}$$

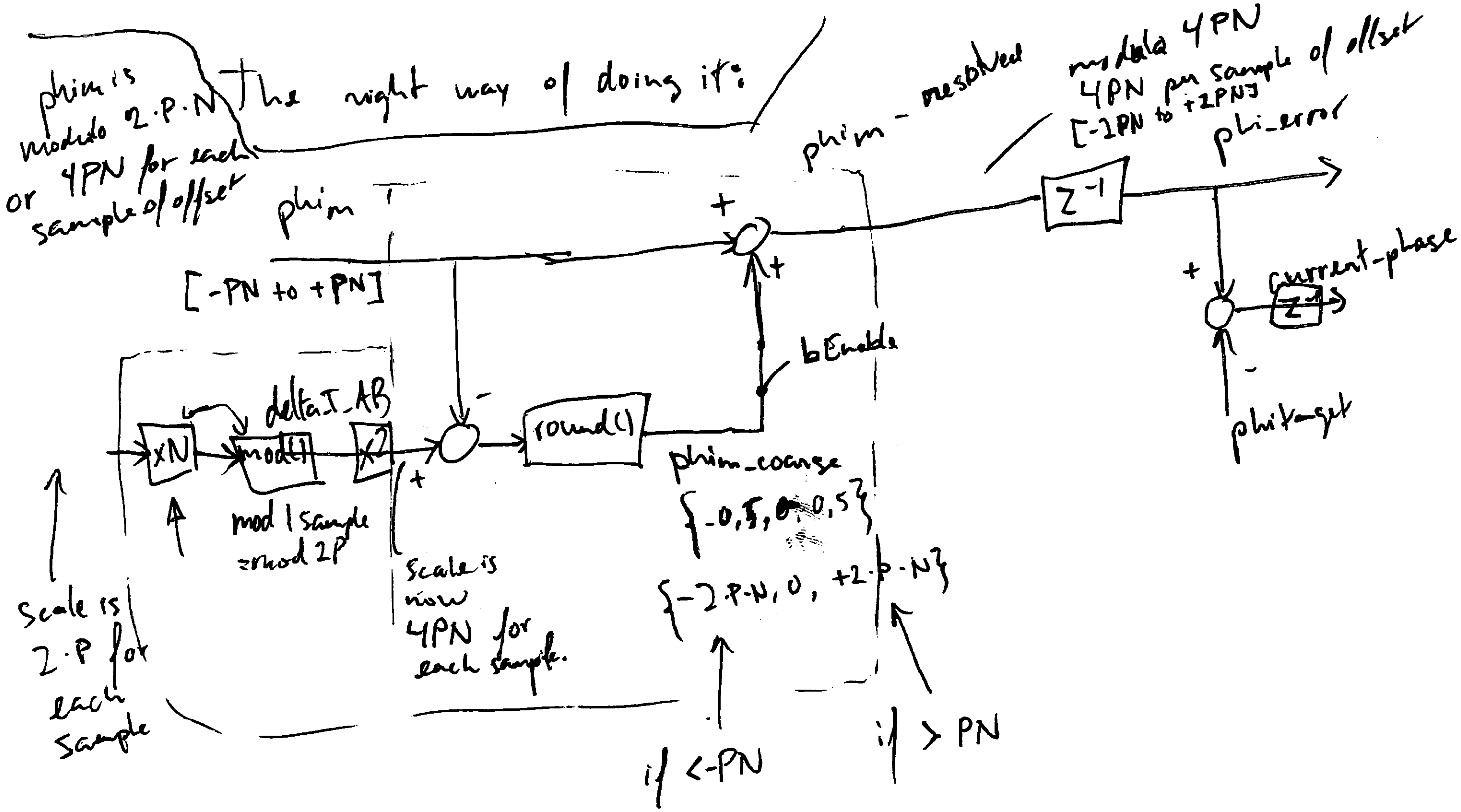
noise that must be smaller than $\frac{x_{modulus}}{2}$

$$x_{coarse} - x_{fine} = x_{noise} - K\, x_{modulus}$$

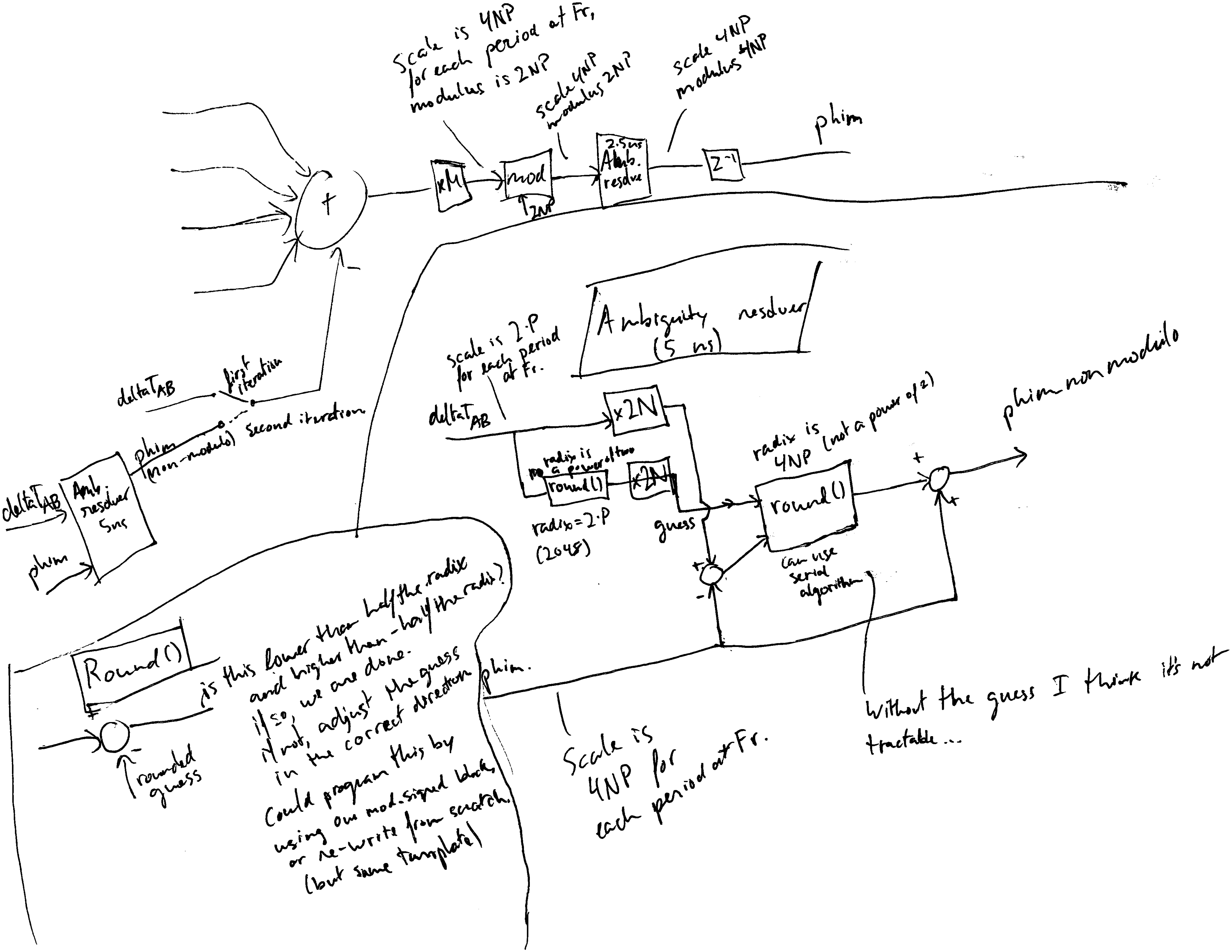
$$\text{round}(x_{coarse} - x_{fine}) = -K\, x_{modulus} \quad \text{as long as } |x_{noise}| < \frac{x_{modulus}}{2}$$

$$x_{fine} + \text{round}(x_{coarse} - x_{fine}) = x_{real} + K x_{modulus} - K x_{modulus}.$$

$$x_{fine} + \text{round}(x_{coarse} - x_{fine}) = x_{real}$$

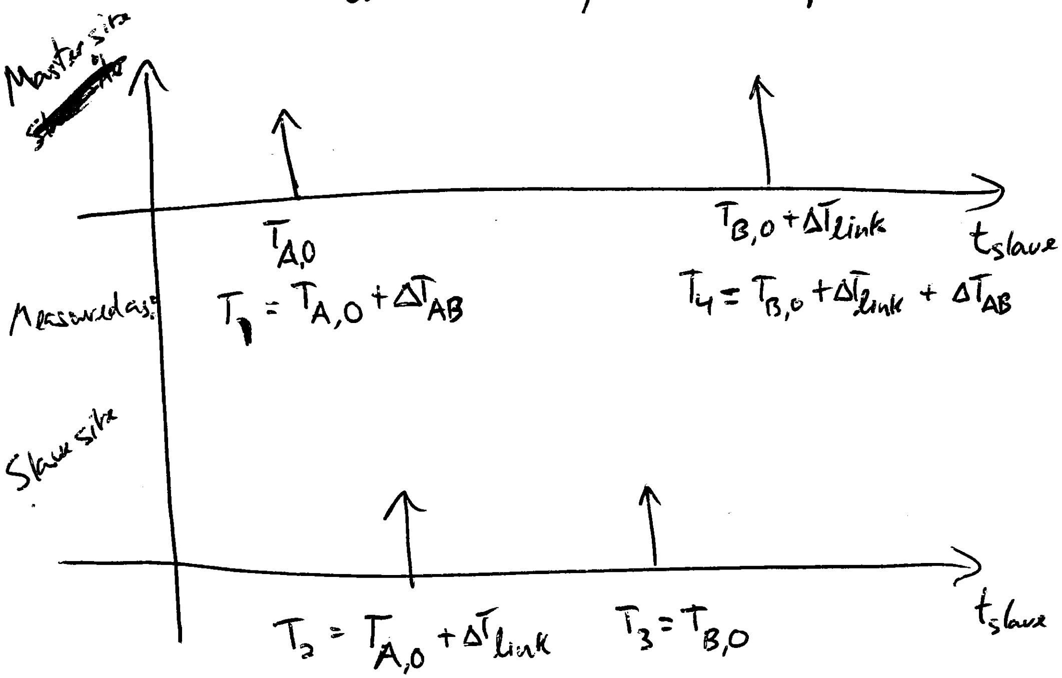


phim is modulo 2·P·N or 4PN for each sample of offset
The right way of doing it:
phim
[-PN to +PN]
phim - resolved
modulo 4PN
4PN per sample of offset
[-2PN to +2PN]
Z -1
phi_error
+
+
current_phase
Z -1
-
phitarget
bEnable
deltaI_AB
xN
mod(1)
x2
-
+
round()
mod 1 sample = mod 2P
Scale is now 4PN for each sample.
phim_coarse
{-0.5, 0, 0.5}
{-2·P·N, 0, +2·P·N}
if <-PN
if > PN
Scale is 2·P for each sample



Scale is 4NP for each period at Fr, modulus is 2NP
scale 4NP modulus 2NP
scale 4NP modulus 4NP
phim
+
-
xM
mod
2NP
2.5 ns Amb. resolver
z^{-1}
deltaT$_{AB}$
first iteration
phim (non-modulo) second iteration
deltaT$_{AB}$
Amb. resolver 5ns
phim
Ambiguity resolver (5 ns)
scale is 2·P for each period at Fr.
deltaT$_{AB}$
x2N
radix is not a power of two
round()
x2N
radix=2·P (2048)
guess
radix is 4NP (not a power of 2)
round()
can use serial algorithm
phim non modulo
+
+
-
Round()
is this lower than half the radix and higher than -half the radix? if so, we are done. if not, adjust the guess in the correct direction.
Could program this by using our mod-signed block, or re-write from scratch (but same template)
rounded guess
phim.
Scale is 4NP for each period at Fr.
Without the guess I think it's not tractable...

Coarse two-way time transfer



"Master site reports timestamps that are ΔT_{AB} too high compared to what the slave would say"

Measurement equations:

$$T_1 = T_{A,0} + \Delta T_{AB}$$

$$T_2 = T_{A,0} + \Delta T_{link}$$

$$T_3 = T_{B,0}$$

$$T_4 = T_{B,0} + \Delta T_{link} + \Delta T_{AB}$$

We want to be insensitive to $T_{A,0}$ and $T_{B,0}$ so we make:

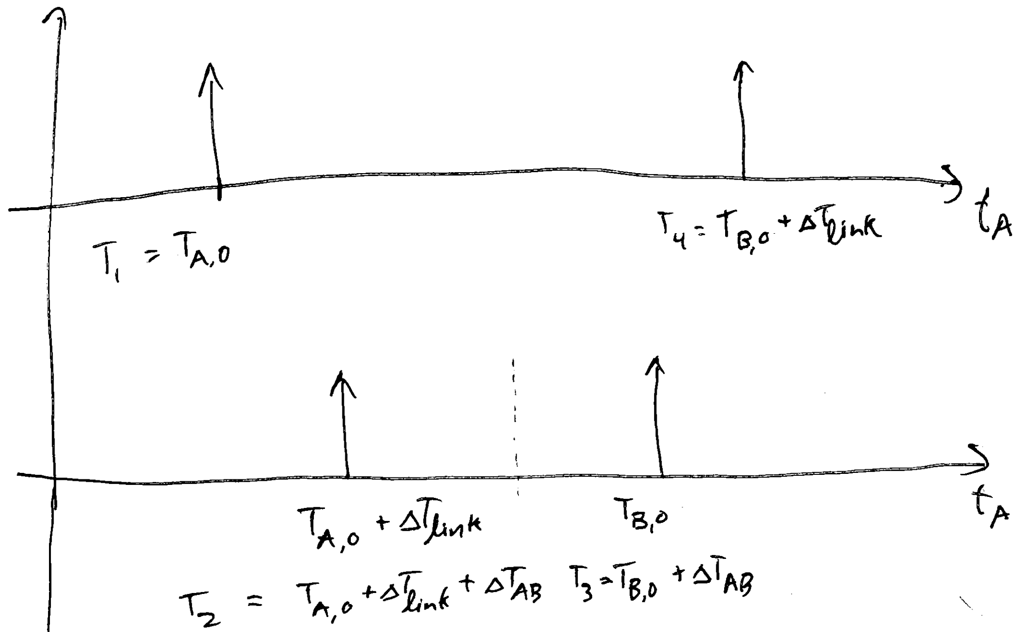
$$T_2 - T_1 = -\Delta T_{AB} + \Delta T_{link}$$

$$T_4 - T_3 = \Delta T_{AB} + \Delta T_{link}$$

Then we isolate ΔT_{link} and ΔT_{AB}:

$$\frac{1}{2}(T_2 - T_1 + T_4 - T_3) = \Delta T_{link}$$

$$\frac{1}{2}\left(T_4 - T_3 - T_2 + T_1\right) = \Delta T_{AB}$$



"Times reported by clock B are ΔT_{AB} higher than would be reported by clock A"

Measurement Equations:

$$T_1 = T_{A,0}$$
$$T_2 = T_{A,0} + \Delta T_{link} + \Delta T_{AB}$$
$$T_3 = T_{B,0} + \Delta T_{AB}$$
$$T_4 = T_{B,0} + \Delta T_{link}$$

We want to be insensitive to $T_{A,0}$ and $T_{B,0}$ so we make:

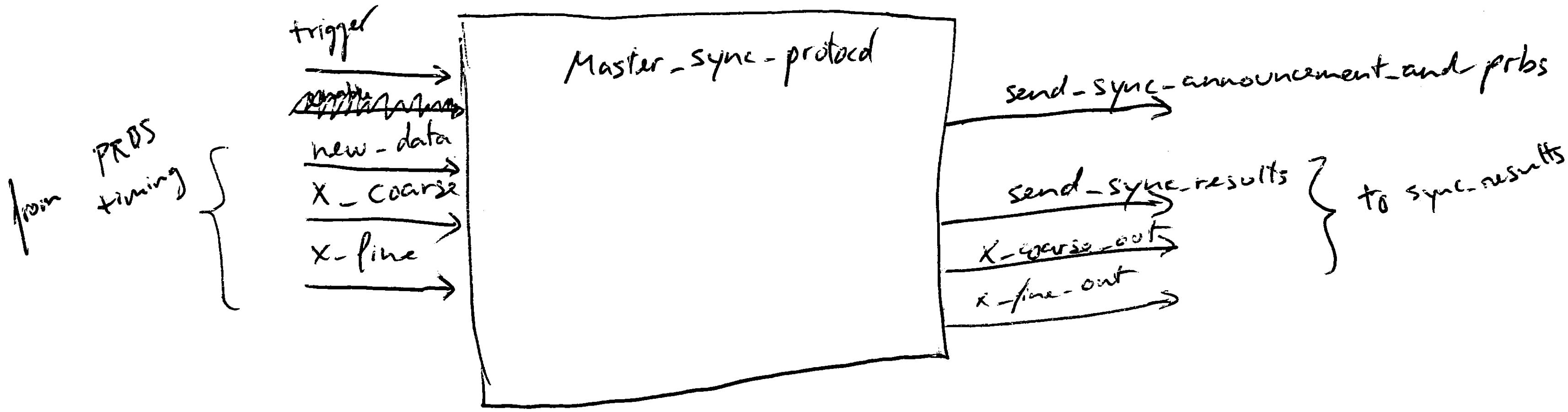
$$T_2 - T_1 = \Delta T_{link} + \Delta T_{AB}$$
$$T_4 - T_3 = \Delta T_{link} - \Delta T_{AB}$$

Then we want to isolate ΔT_{link} and ΔT_{AB}:

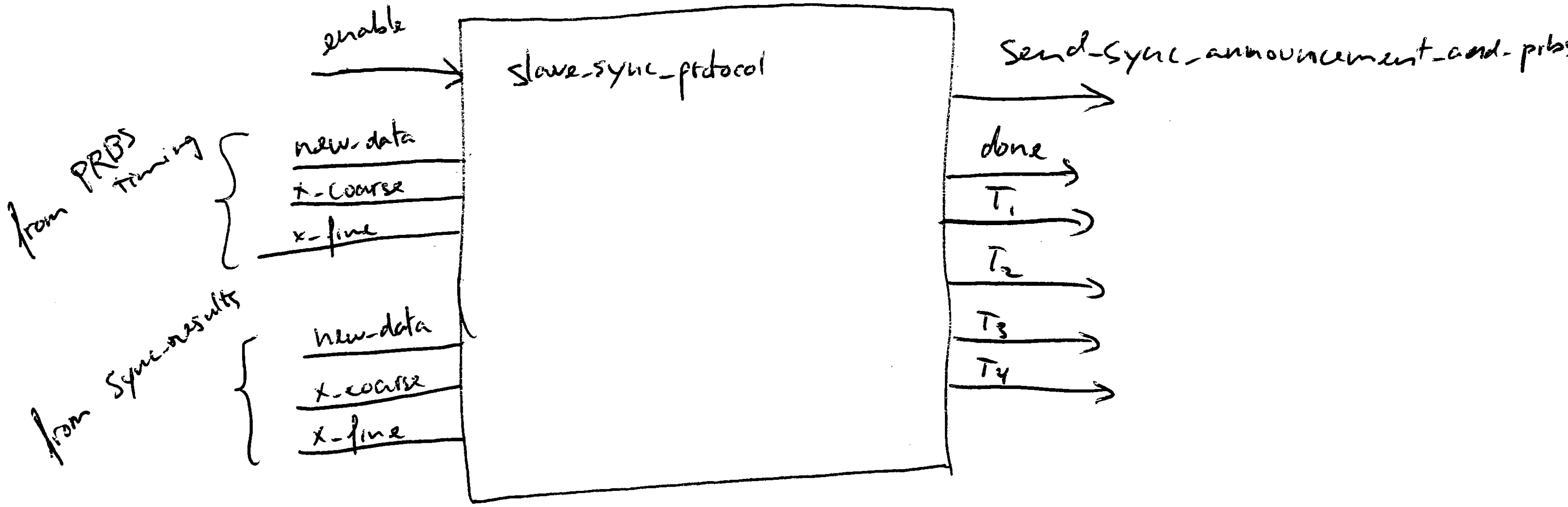
$$\frac{1}{2}\left(T_2 - T_1 + T_4 - T_3\right) = \Delta T_{link}$$
$$\frac{1}{2}\left(T_2 - T_1 - T_4 + T_3\right) = \Delta T_{AB}$$

→ Anything that affects all four measurem[ents] equally cancels out.

→ The effective measurement time for Δt_{AB} is $\frac{1}{2}\left(T_{A,0} + \Delta T_{link} + T_{B,0}\right)$, which $T_{B,0}$ should be (almost) deterministically related to $T_{A,0} + \Delta T_{link}$. Jitter should be around ± 1 bit sample time of the comm link.



Master_sync_protocol
trigger
new_data
X_coarse
x_fine
from PRBS timing
send_sync_announcement_and_prbs
send_sync_results
X_coarse_out
x_fine_out
to sync_results



slave_sync_protocol
enable
from PRBS timing
new_data
x_coarse
x_fine
from sync_results
new_data
x_coarse
x_fine
send_sync_announcement_and_prbs
done
T1
T2
T3
T4

How can the PRBS TWTFT give $1.5\Delta T_{link}$ sometimes?
ΔT_{AB} is just fine when that happens

$$\frac{1}{2}\left(T_2 - T_1 + T_4 - T_3\right) = \Delta T_{link}$$

$$\frac{1}{2}\left(T_4 - T_3 - T_2 + T_1\right) = \Delta T_{AB}$$

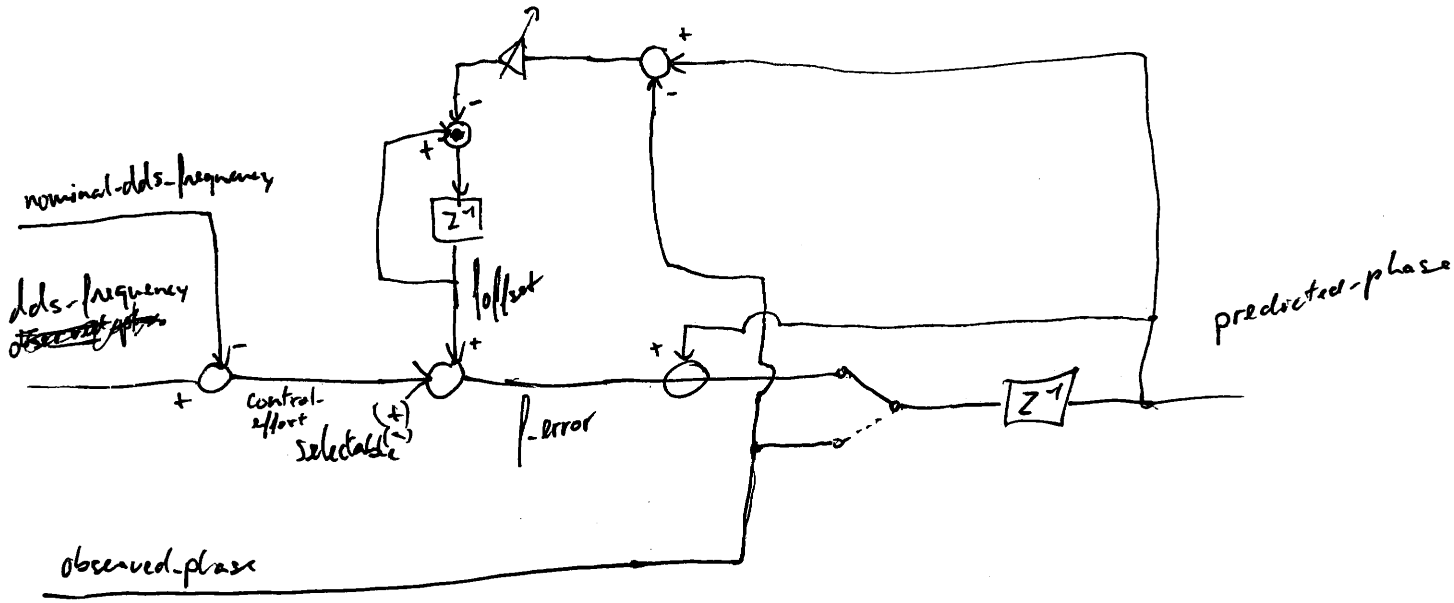
$$T_1 = T_{A,0} + \Delta T_{AB}$$

$$T_2 = T_{A,0} + \Delta T_{link}$$

$$T_3 = T_{B,0}$$

$$T_4 = T_{B,0} + \Delta T_{link} + \Delta T_{AB}$$

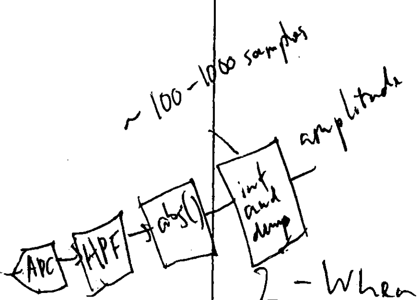
Observer/Linear predictor



the combination of the three phases

Calibration mode:

1 - First start by locking the phase to 0. This can only start when we have: - local IGM phase
- Remote IGM1 phase
- Remote IGM2 phase
- Coarse sync protocol complete:
  - delta_T_link
  - delta_T_AB



2 - When we get the signal from the user, we start slowly ramping the ref phase, eg at 1 or 0,1Hz. While this is happening, we run a max finding algorithm on the validation IGM amplitude, saving the peak amplitude and the ref phase at that peak. BTW (Data transmission, sync protocol all need to run while this is happening).

3 - Either (when the user says so) or (after 2π phase shift of the ref phase), I am no sure which criterion to use, set the ref phase to equal the ~~peak value~~ one where we have seen the peak interference.

At this point we should be synchronized: the validation IGM shows a tone with the strongest amplitude

4. Maybe here we could allow the user to manually add an offset to ref phase, in case the automatic peak finding gives too much bias. It could also be nice to have a mode where the ref phase ramps up and down over a narrow range.

5. Once the user is satisfied that the synchronization is good, the calibration is finished. The calibration set comprises:
- Ref phase : Calibrates mostly receivers' internal delays
- Matched filters for both sites
- Delta_T_link : To calibrate the offset between the link delay seen by the comm link vs the one seen by the dual-comb setup.
- Delta_T_AB : To calibrate essentially the ~~delay~~ fractional delay between the sample clock and the optical pulses (for absolute timing)
- Also is necessary to resolve the 2.5ns ambiguity because the ref phase is modulo 2π and relates to the two-way phase, effectively halving the non-ambiguity range.

Normal Operation mode:

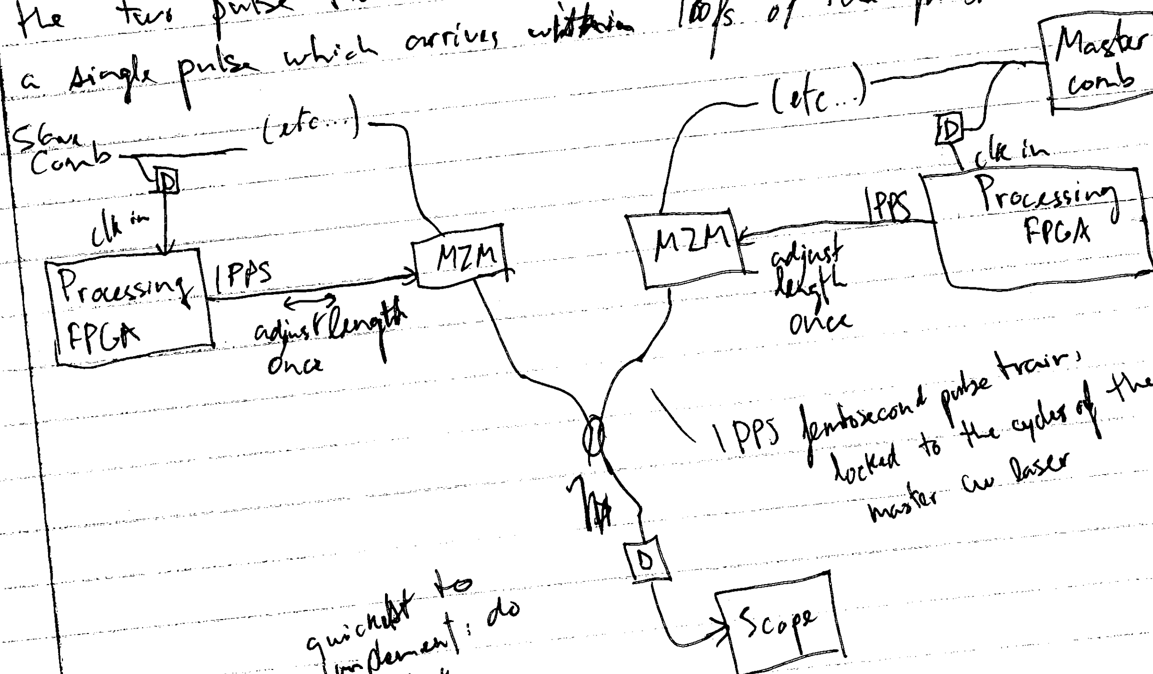
1. Generate the measurement phase by combining:
   - Local IGM phase
   - Remote IGM 1 phase (transfer - slave), adjusted by delta_T_AB
   - Remote IGM 2 phase (transfer - master), adjusted by delta_T_AB
   - $2\pi f_r$(delta_T_link - delta_T_link_calibration)
2. Lock the measurement phase to ref_phase_calibration.
3. To resolve the 2.5ns ambiguity, we need to look at the fractional part of (delta_T_AB - delta_T_AB_calibration) when the pulses are locked. This system should arbitrarily lock to either 0 or ±0.5, at which case we should add a 2π shift to the measurement phase. Note that this seems the simplest way of doing things initially but there probably is a way to adjust the lock point at first instead of waiting for the pulses to be locked and then figuring out if we were right or wrong.
   I think that we simply need to compare the measurement phase and the fractional part of (delta_T_AB - deltaTAB_calibration), appropriately scaled.

To get absolute time sync: Let's call this a "1PPS-like" signal?

- We simply need to output a 1PPS signal from both FPGAs,
at the master site: free running.
slave site: output a pulse according to timestampcounter minus (or plus, I am not sure, but I think ~~its~~ minus) delta_T_AB.

Note that we should

- Manually adjust the delay between this 1PPS output to always be centered on the optical pulse train. This signal could drive a modulator on the way to the validation coupler which combines the two pulse trains. This way the 1PPS could be used to "paint" a single pulse which arrives ~~within~~ 100fs of the master.



quickest to implement: do this first

$1PPS$ vs $\frac{F_r}{2^{K_1}}$ vs $\frac{F_r}{K_2}$

↳ cleanest, but how the hell do we do that?

$F_r = \frac{F_{cw}}{N}$, and we can then easily do $F_{out} = \frac{F_r}{K_2}$, but then gives: $F_{out} = \frac{F_c}{K_2}$

So this only becomes 1PPS if $\frac{F_{cw}}{1Hz}$ is an integer, $K_2 \cdot N$ (and a specific one)

UI for Syntonization

Blue FPGA

~~Initial setup~~ | Initialization | (ADC, Ethernet, constants)

Setup ADC and ethernet

Opens graphs / ~~####~~ Grab and save matched filter to disk, make sure N=25 threshold:

Load MF from disk and set to current ~~Set matched filter~~ to current — Opens graphs

• Blue
• Orange

Opens graph — Grab noise statistics

Setup triggering on $Abs()^2$ signal ~~Pea~~

Threshold:
- From stats
- From stats, with add'l margin
- Nominal peak/10

Opens graph — Grab single dataset | Grab & display |

T [s] | 1 | | Abs | | Display last |

Spawn UDP receiver process

Continuous display (one-way)

Continuous display (two-way) (timer) | ⊙ Off | ⊙ On |

Move up before grabbing data — Lock comms beat frequency | ⊙ Off | ⊙ On |

~~Freq offset~~ Initial setpoint [Hz]

Transmit data over comm link | ⊙ Off | ⊙ On |

~~Auto coarse~~

IGM triggers coarse sync ☑

Coarse sync protocol: ⊙ Slave ○ Master

Orange FPGA

Same, but also with:

Syntonization

Local modulus: 100e3 (use eval())

Remote modulus: 100e3-1

Lock mode:

⊙ Lock Off, Reset error ~~to zero~~ ⊙ Lock off, no reset

○ Lock On, no reset ~~Reset~~ ⊙ Lock on, reset

Loop tuning

☑ Two-way (changes gain only)

Desired BW [Hz] | 100 |

Sign
- Positive
- Negative

Nominal DDS freq

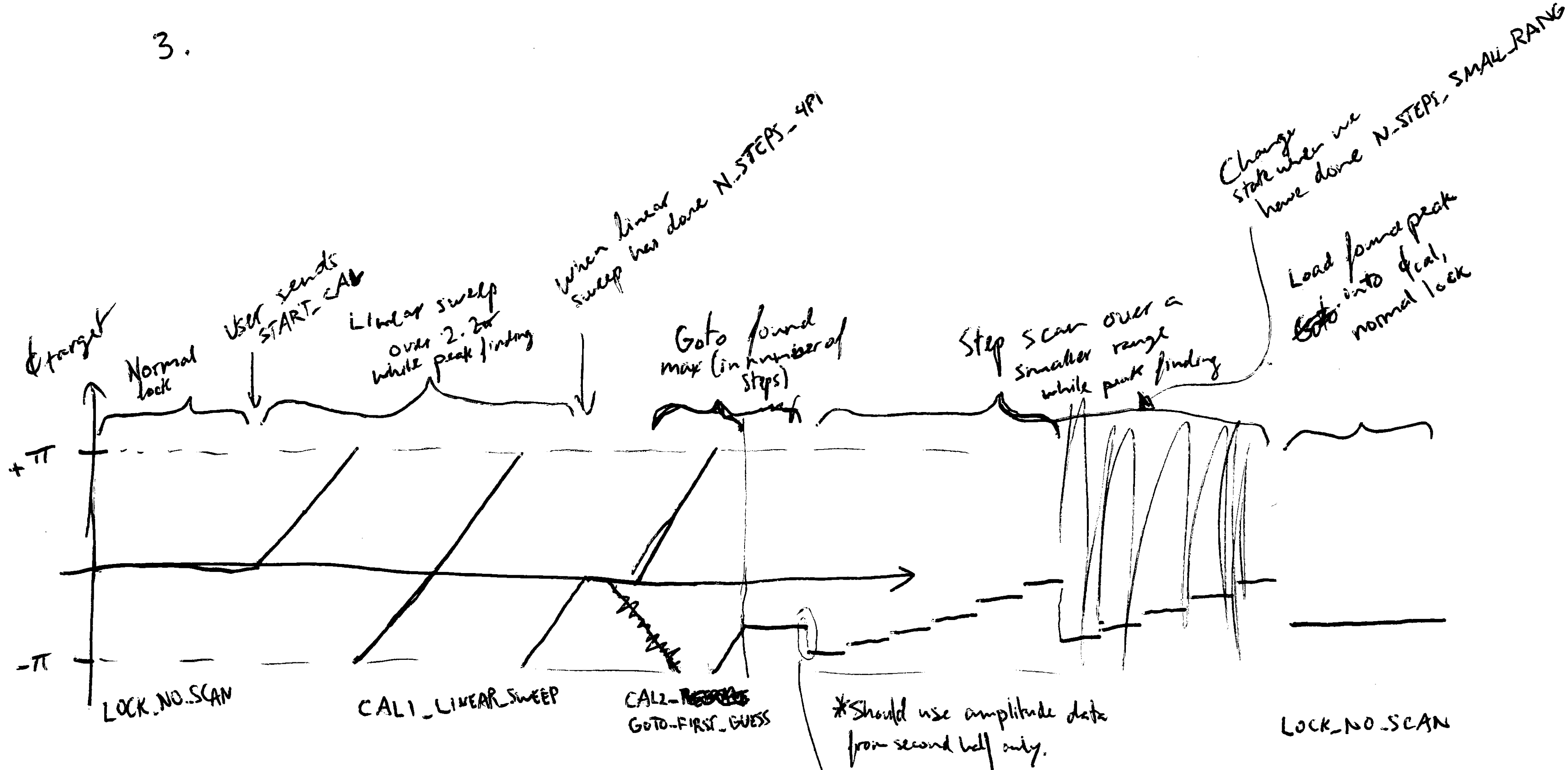
| 45.0 | MHz | Update |

1. Lock to 0 (no scan)

2. Start linear scan (scan, isStep=0 ~~stepsize linearsteps, waittime~~ ...)

3.

Idea for calibration procedure (automated?)



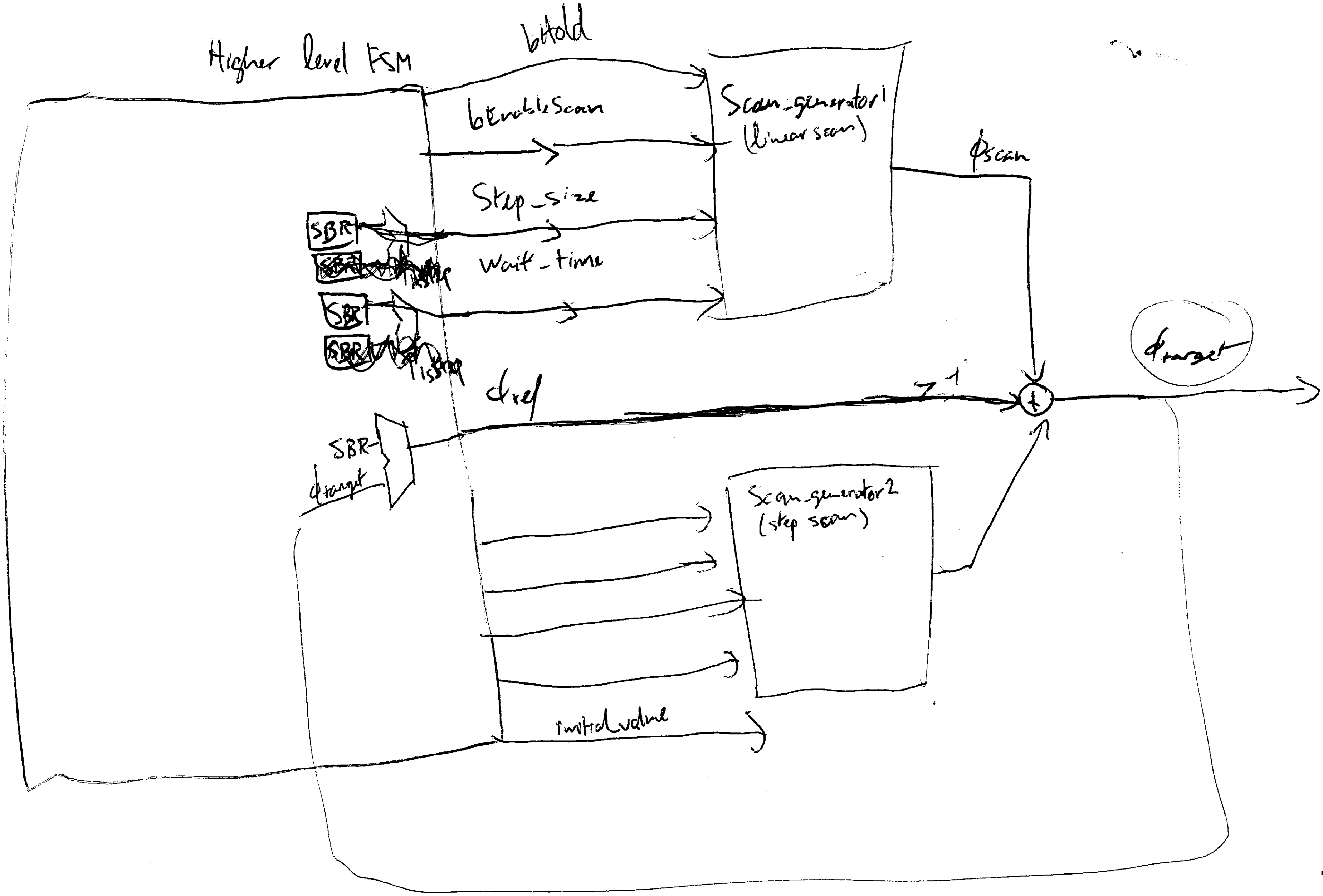
We also need a mode which activates the two scanning modes without going through the calibration states

LOCK_LINEAR_SWEEP

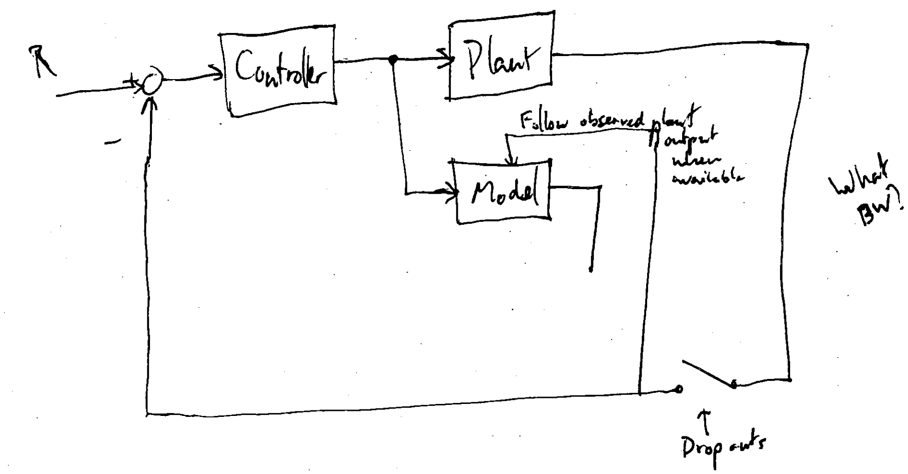
LOCK_STEP_SCAN

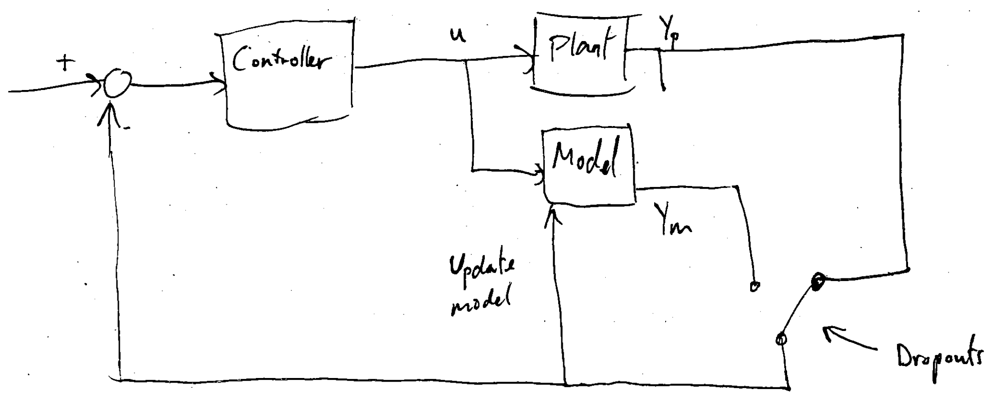
Δ Need to be able to put an initial value to the step scan.

CAL3_STEP_SCAN


Higher level FSM
bHold
bEnableScan
Step_size
Wait_time
Scan_generator1
(linear scan)
ϕscan
ϕtarget
ϕref
SBR
isStep
SBR
ϕtarget
Scan_generator2
(step scan)
initial value

Trying to figure out
how to do closed-loop/open-loop transition (when there is a dropout)





→ We observe position (phase)

→ The observation gets turned on/off by turbulence.

→ The real drifts of the plant are mostly slowly changing linear frequency drifts.

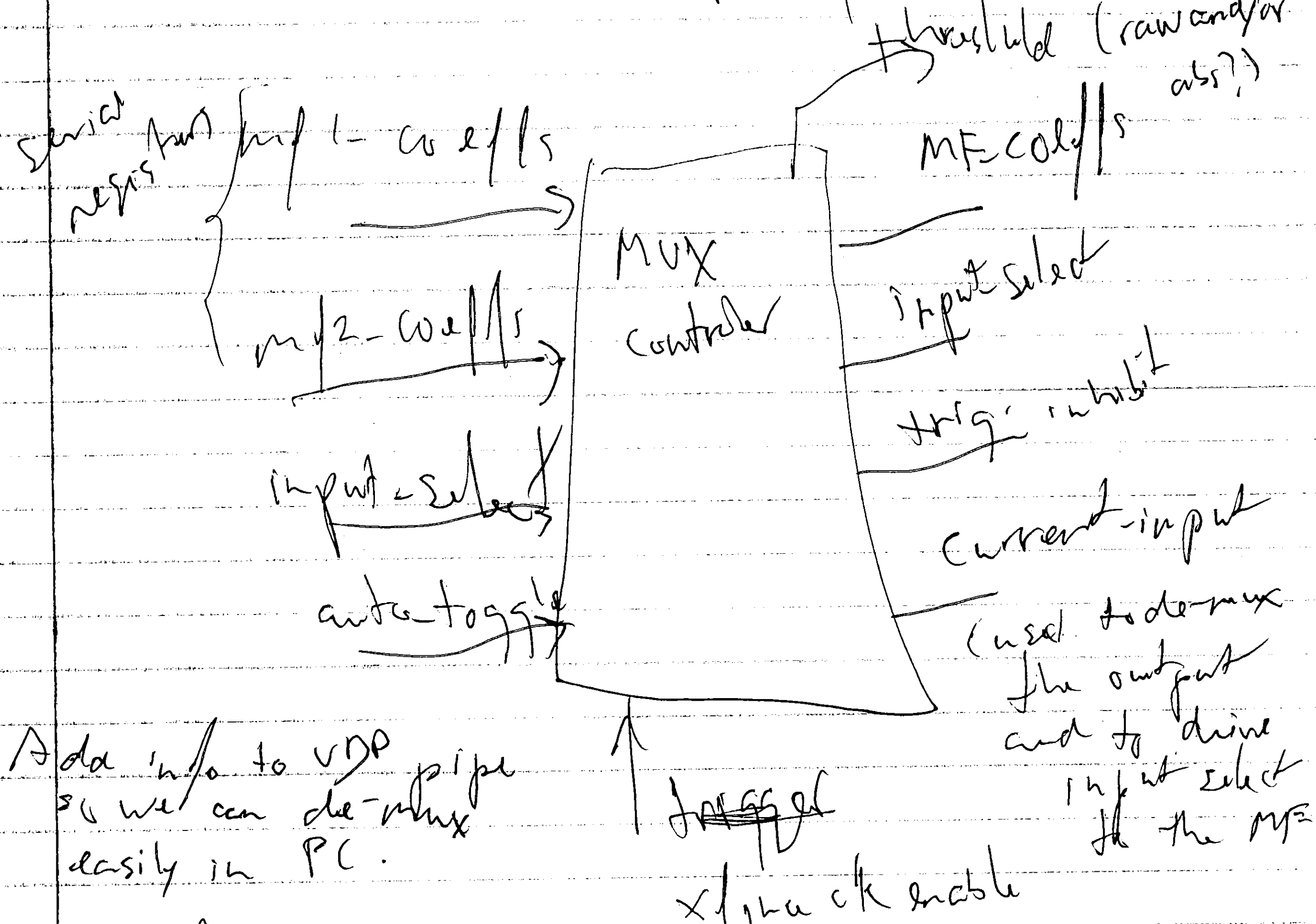
→ We have white measurement phase noise.

→ The two-way makes things slightly more complicated by adding delay to half of our response

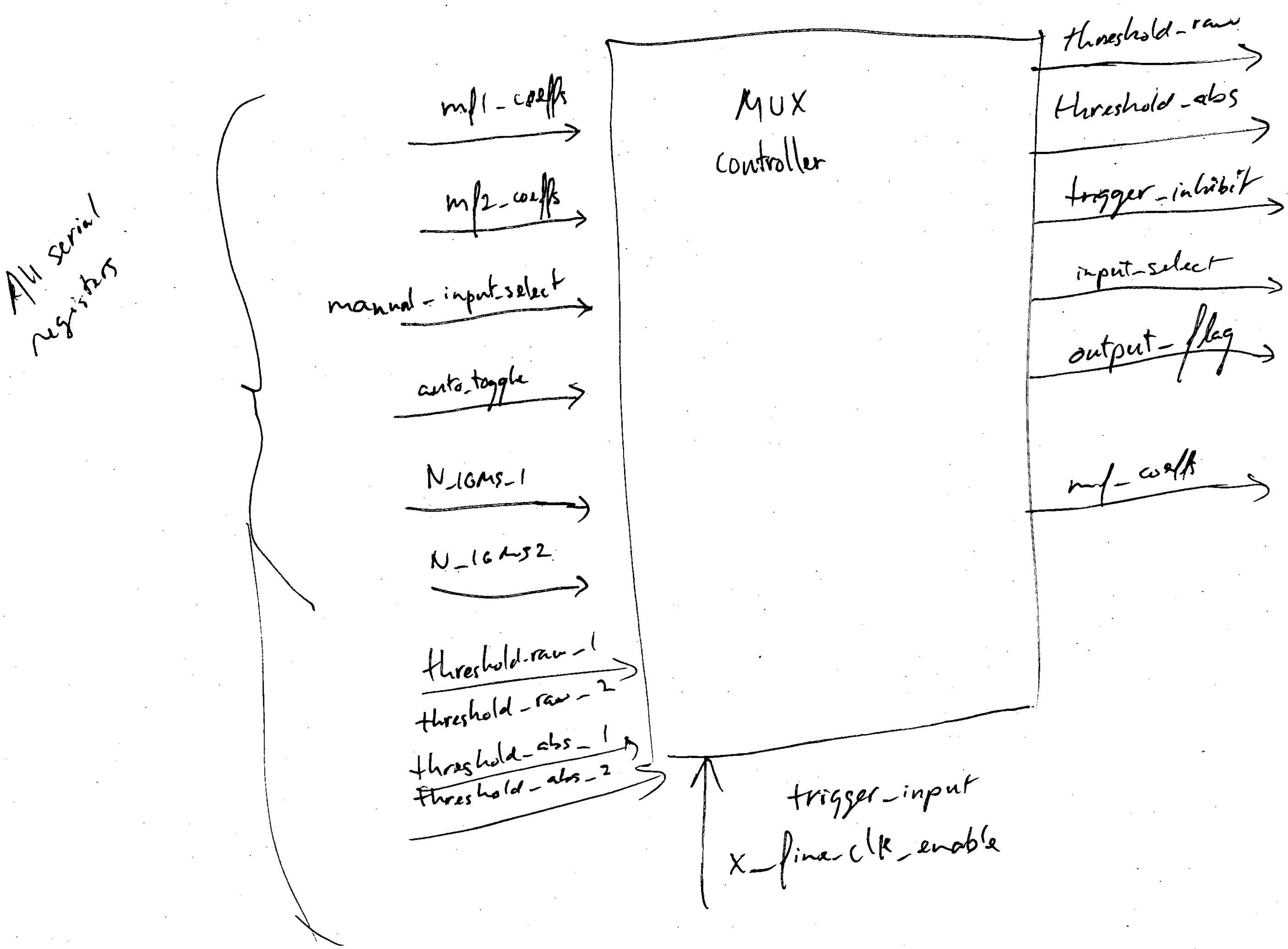
→ We know the system dynamics really well. (Pure integrator with very accurately known gain).

Preliminary design of Mux controller

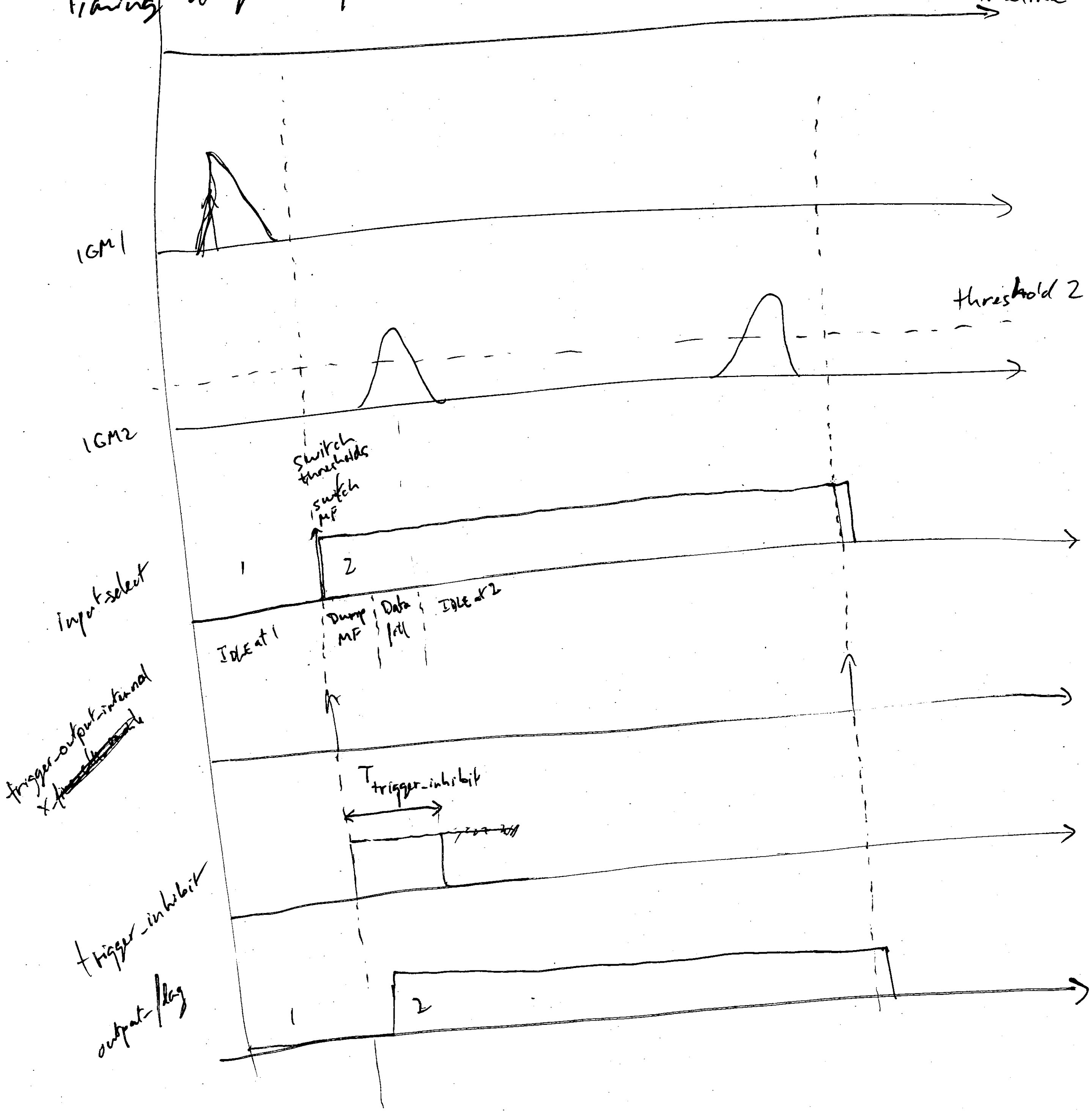
- Muxing of sgn processing.
  - Add trigger ~~+~~ inhibit input/state to the triggering logic. Only comes out when trigger inhibit = 0, value below threshold.
- Add a several HPF
  - MUX between output of each HPF



- Add info to UDP pipe so we can de-mux easily in PC.
  - New output packet type
  - What about UPP? Just a flag!
- Delay of the two Matched filters have to be set equal by MATLAB so we don't have to change this tool.


All serial registers
mf1_coeffs
mf2_coeffs
manual_input_select
auto_toggle
N_IGMS_1
N_IGMS2
threshold_raw_1
threshold_raw_2
threshold_abs_1
threshold_abs_2
MUX controller
threshold_raw
threshold_abs
trigger_inhibit
input_select
output_flag
mf_coeffs
trigger_input
x_fine_clk_enable

Timing diagram for MUX controller



$T_{trigger\_inhibit}$ needs to be long enough to change the whole MF, and then fill it with valid data from the other input, plus fill the peak finder taps with valid data. So I think at a minimum:

$$\underbrace{200}_{\text{MF switch}} + \underbrace{200}_{\text{MF fill}} + \underbrace{10}_{\text{Abs()}^2\text{ fill}} + \underbrace{50}_{\text{peak finder fill}} = 460$$

State machine for MUX controller

icm_detected is either trigger_on_row or x_line_clk_enable depending on trigger_select