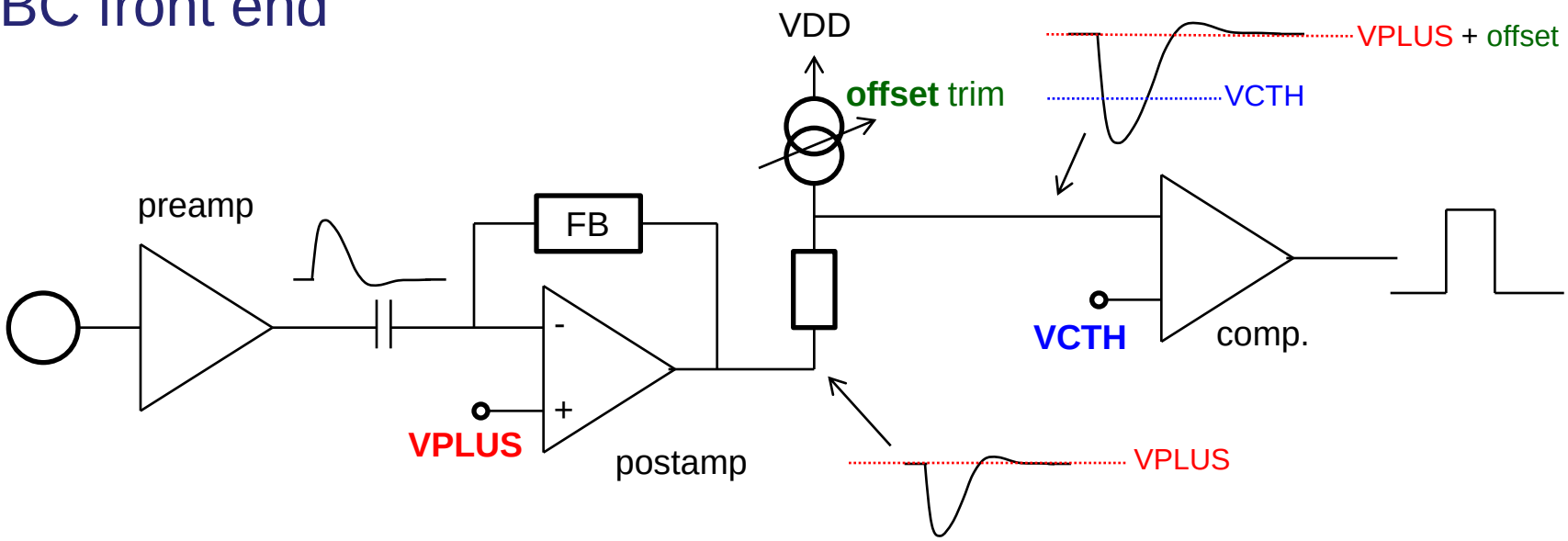
CBC front end



3 voltage levels to program

VPLUS sets DC point at output of postamp
offset current adds an extra +ve DC shift
VCTH is the comparator threshold

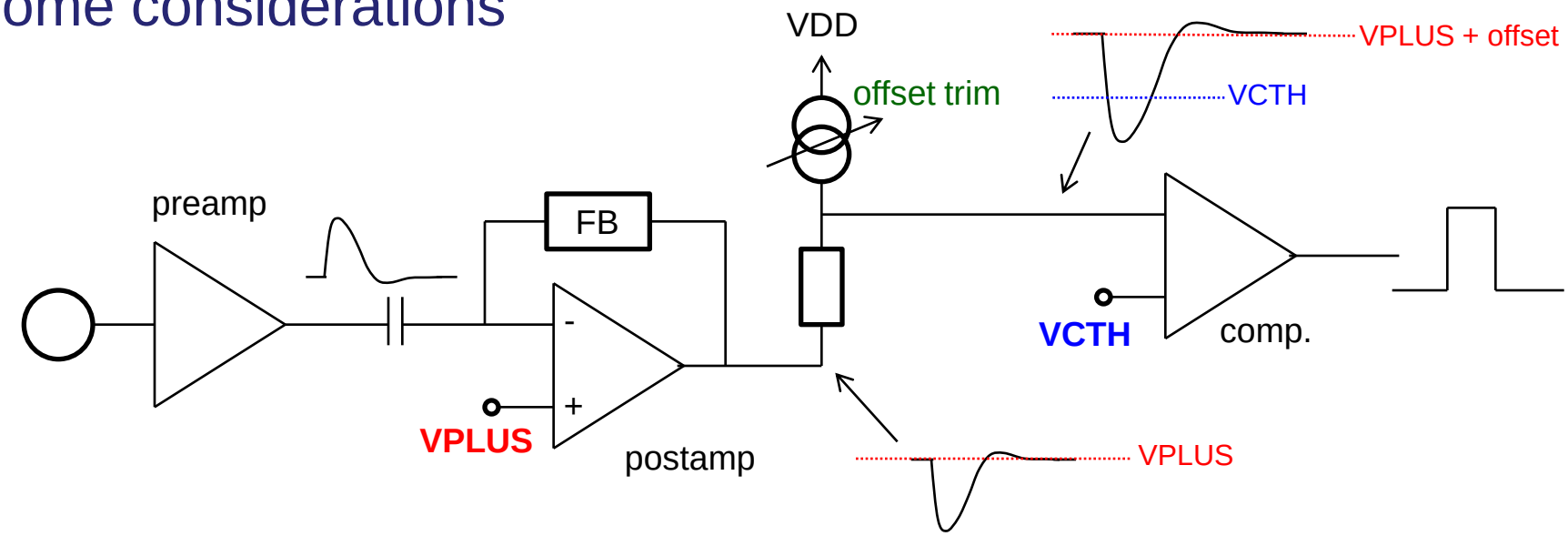
VPLUS, **VCTH** are global (same value for all channels)
offset current is individually adjustable for every channel

what should we choose for **VPLUS**? what is the best strategy for tuning offsets?

have given it some thought and would like to propose a “standard” strategy

developed on custom test-bench, now coded for GLIB based DAQ

some considerations



VPLUS and **VCTH** generated in the same way, so ~same adjustment range

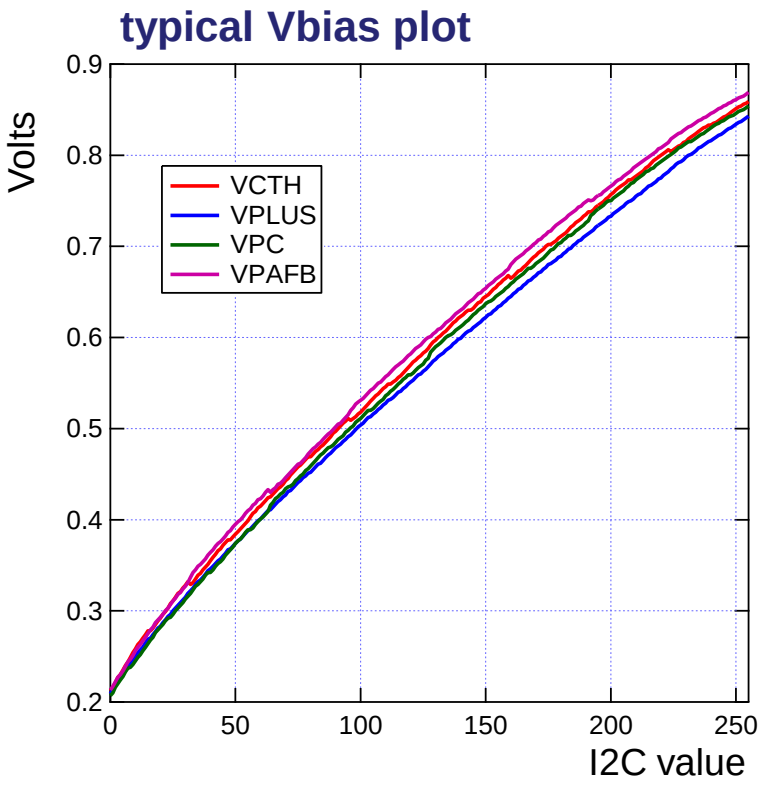
VPLUS value should allow acceptable dynamic range for signal

i.e. not too close to low end of range for electrons mode as signal excursion will be -ve (somewhere in the middle of the power supply range is ok)

offset current should not be too close to zero

some current needed to avoid slew-rate limiting at comparator input node

what do the on-chip voltage biases look like?



all voltages generated in same way

=> range of adjustment ~ same (some chip-to-chip variation)

not perfectly monotonic - discontinuities apparent
at major binary boundaries (32, 64, 128)

monotonicity varies chip-to-chip

only matters for VCTH - will seek to improve for CBC3

resolution ~ 2.5 mV / I2C unit => ~ 300 electrons

probably ok but may also seek to increase
resolution for CBC3

monotonicity is the main issue - why?

affects precision of threshold setting, particularly for a channel where the offset tuning point
is at one of the significant discontinuities

old s-curve acquisition strategy for offsets tuning

- ultimate goal is to bring all channel pedestals into alignment

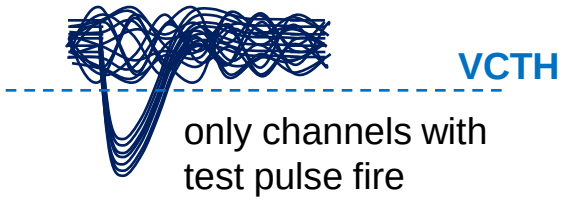
  pedestal => absence of signal (=> no test pulse)

- but up to now approach has been to tune offsets with small test pulse signal

  legacy of expecting to operate with hit-detect circuit

  has advantage that only fire channels corresponding to test pulse group selected

  i.e. all channels **not** firing at once



new s-curve acquisition strategy for offsets tuning

- more likely mode of operation now with comparator continuously sensitive (hit-detect off)

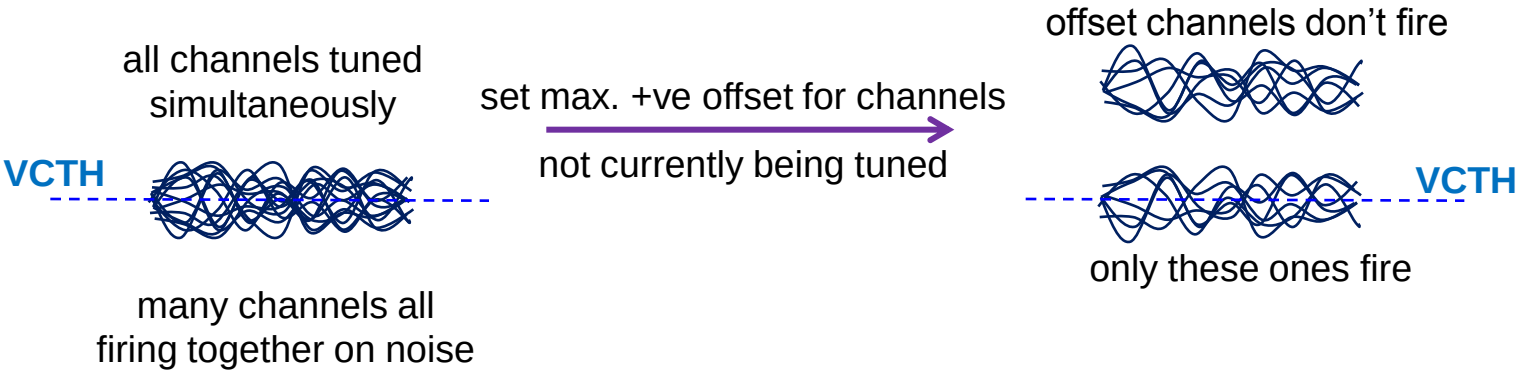
  so can acquire s-curves just by sweeping comparator threshold without test pulse

- can avoid all channels firing at same time by programming offset to channels not being tuned

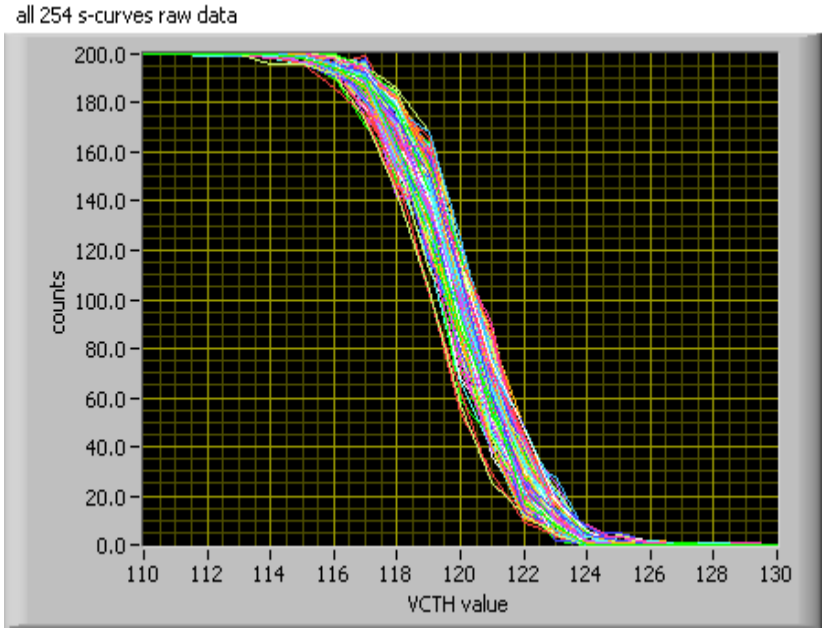
  => keep size of group of channels being tuned to 32 (same as test pulse group)

  (in principle could be any size you like)

- this is the approach we are now recommending to use



effect of all channels firing at once

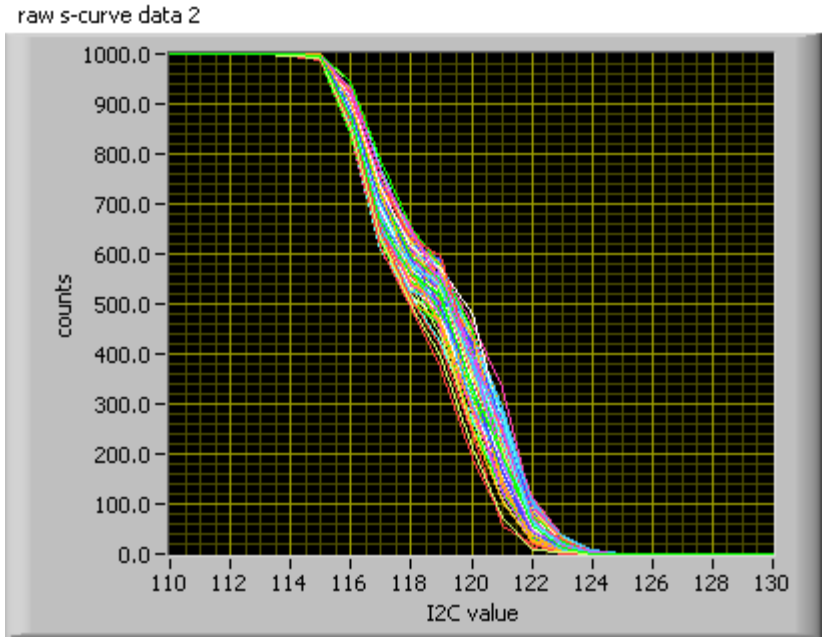


all 254 s-curves present in both plots

acquired in groups of 32

other channels offset so not firing

s-curves look normal



all 254 acquired simultaneously

all channels firing at once

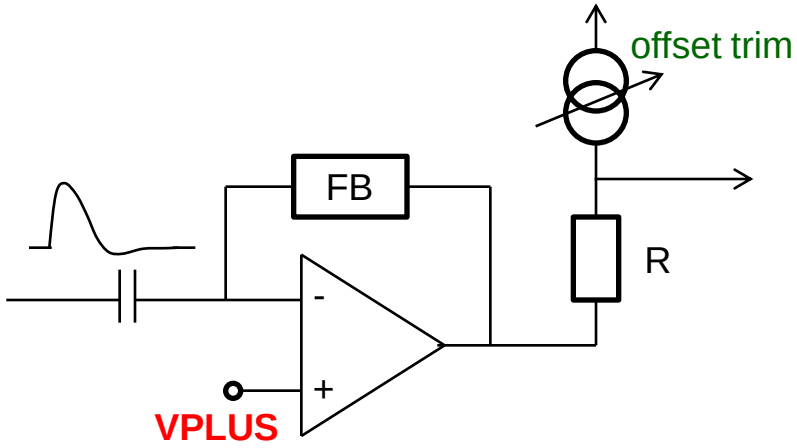
s-curves show distortion

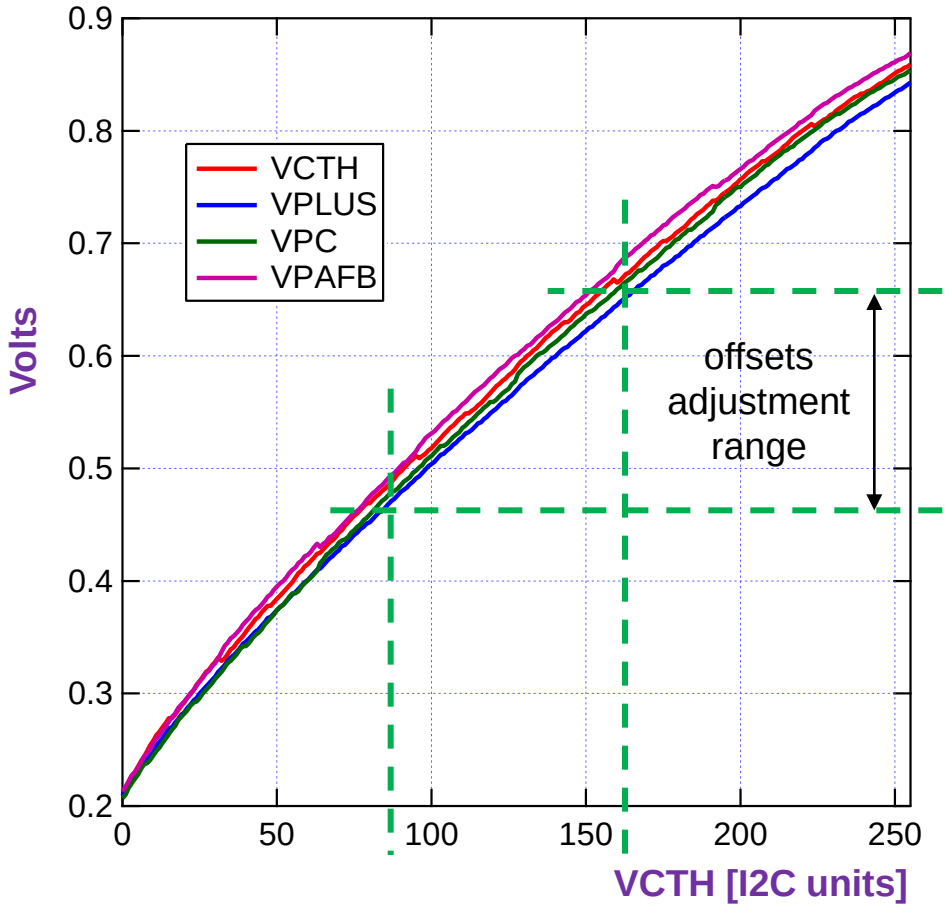
must be due to major digital activity
as threshold passes through channel pedestals

offsets tuning

individual channel offset depends on:
fullscale offset current ~ 10 uA
resistor value ~20k

=> adjustment range ~ 200 mV





spread of final offset values will depend on matching of transistors, offset currents and resistors across the chip

can study by looking at offsets distribution after tuning for range of target s-curve midpoints

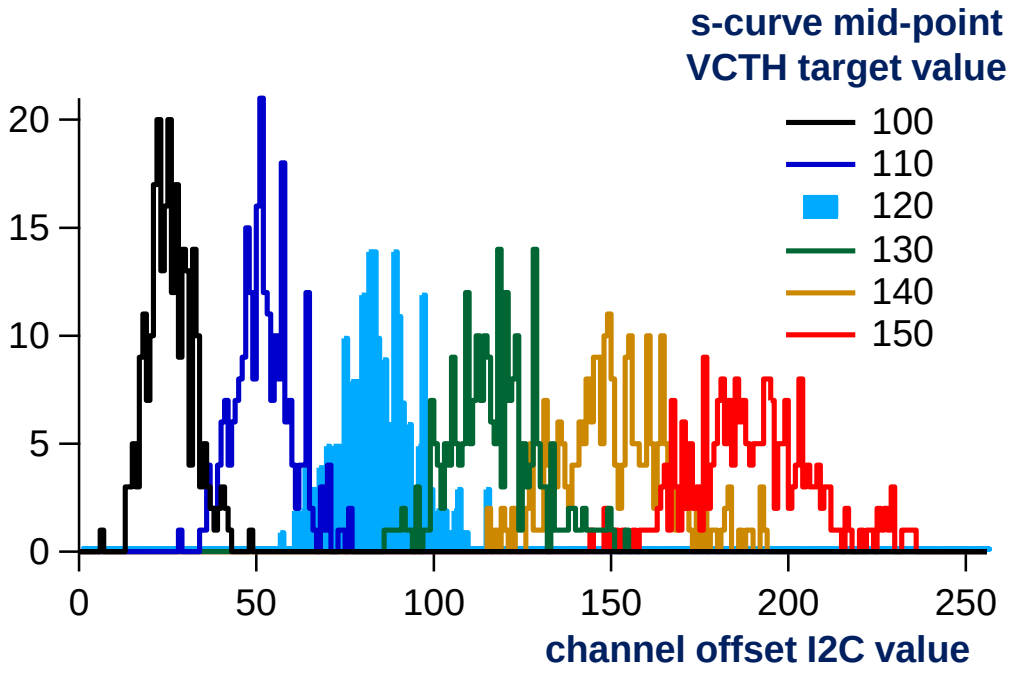
note: s-curve mid-point = pedestal (if no signal)

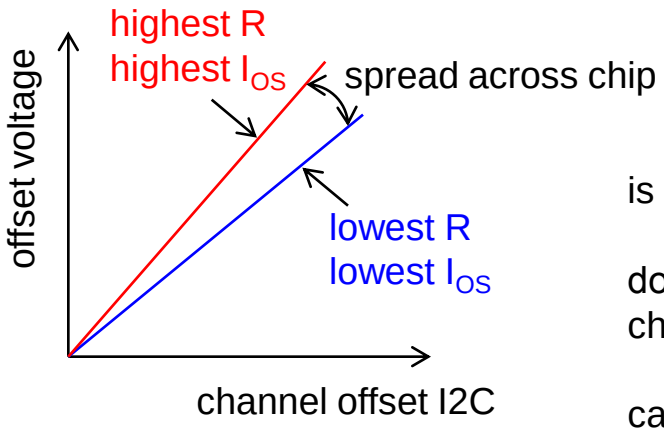
offset spread as function of tuning point

histogram shows distribution of I2C offset values resulting from tuning to achieve different s-curve mid-points

offsets spread increases with target value

likely due to across chip spread in current DAC and offset resistor values





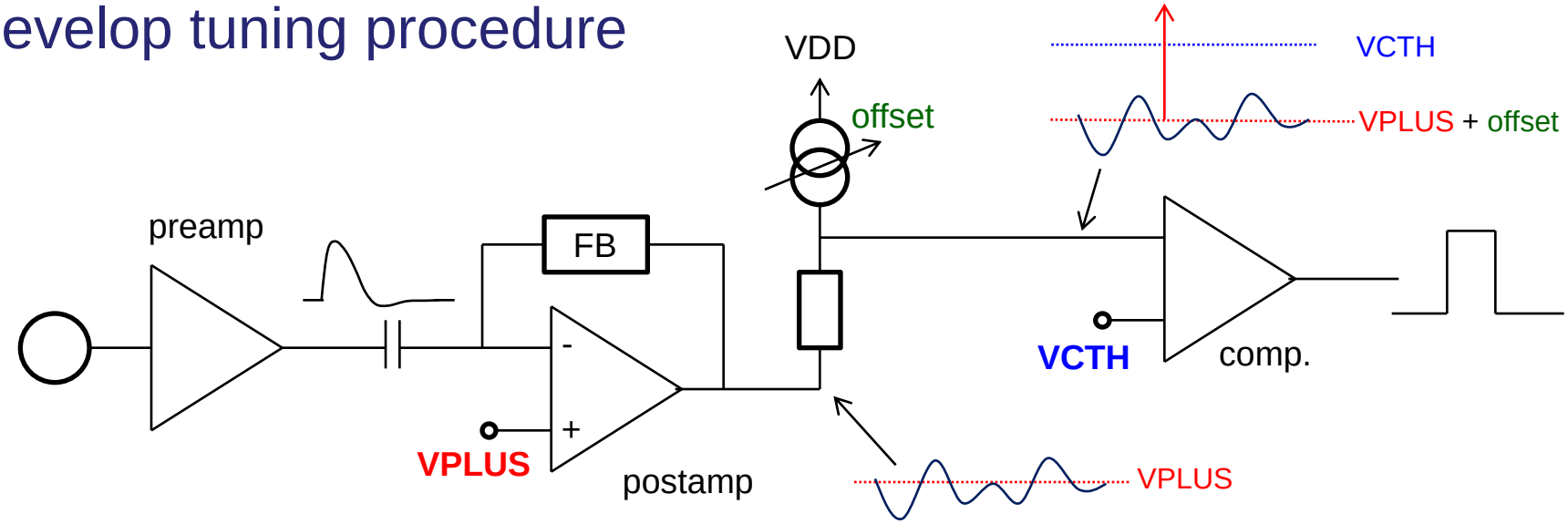
is this a problem?

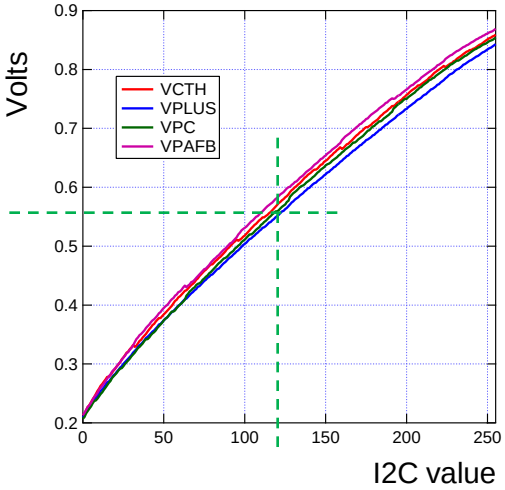
don't see why, but should probably aim to get average channel offset somewhere in the middle of the range

can ensure this by appropriate choice of VPLUS

e.g. let's choose to aim for average channel offset value of ~80

develop tuning procedure





- don't want offset values too small (slew-rate)

  start by setting all offsets to 80

- want channel pedestals somewhere in middle of power supply range

  set VCTH to target value of 120 (~ 0.55 V)

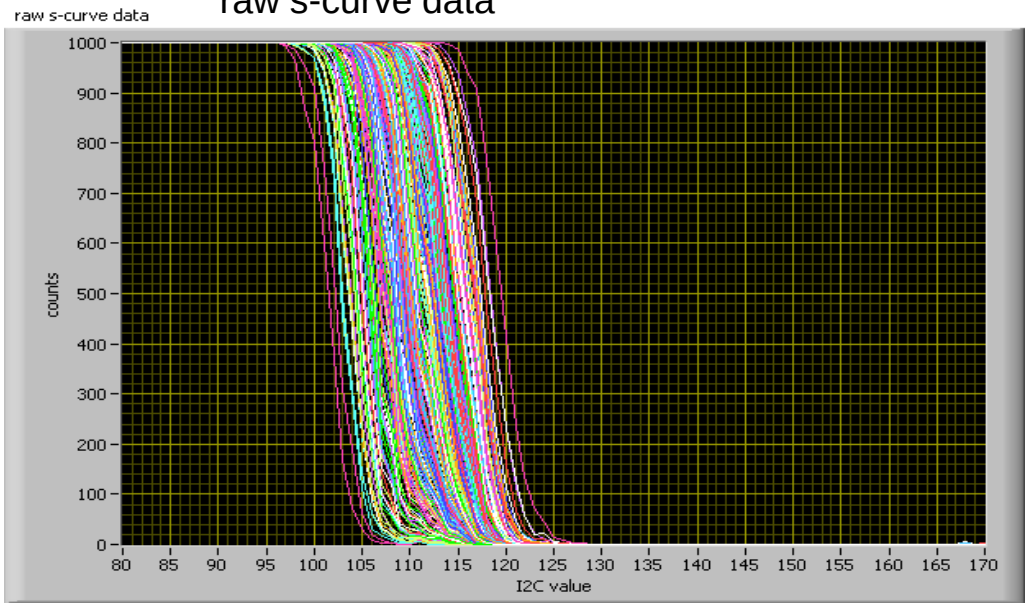
- now acquire a set of s-curves by sweeping VPLUS

- goal is to find the value of VPLUS where (on average) half the channels are firing

  i.e. the average VPLUS s-curve mid-point value

VPLUS tuning procedure example

raw s-curve data



VCTH set to 120_d

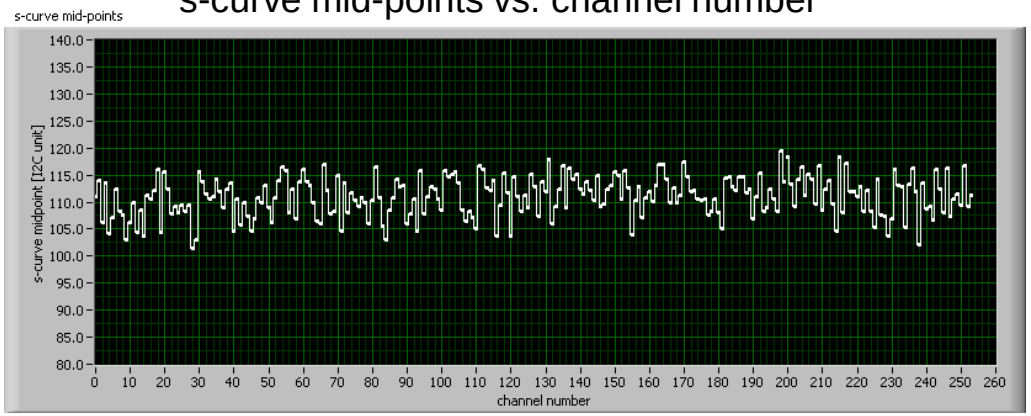
all individual channel offsets set to 80_d

s-curves acquired

note: sweeping VPLUS (**not** VCTH)

VPLUS I2C value

s-curve mid-points vs. channel number



average s-curve mid-point calculated

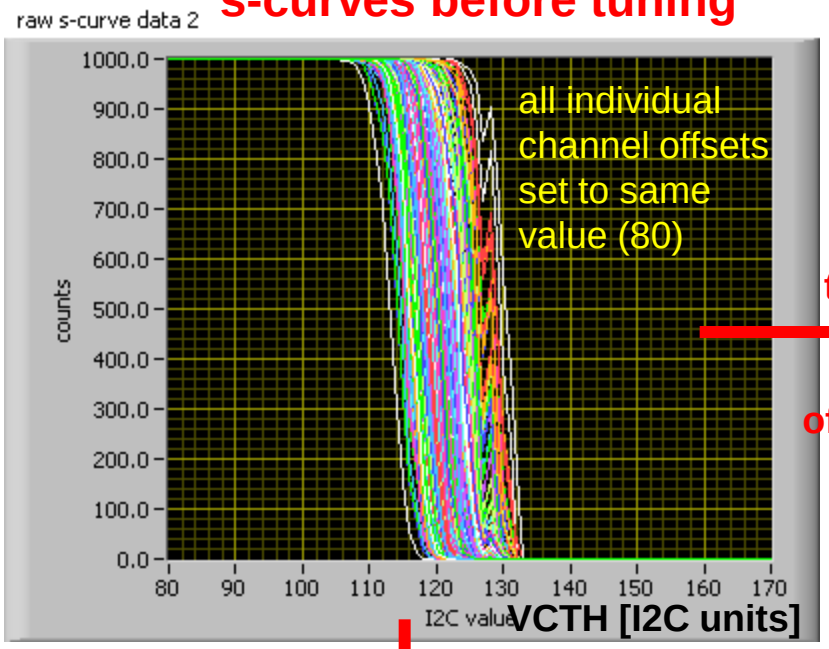
~ 111_d for this chip

VPLUS value can now be fixed at 111_d

channel no.

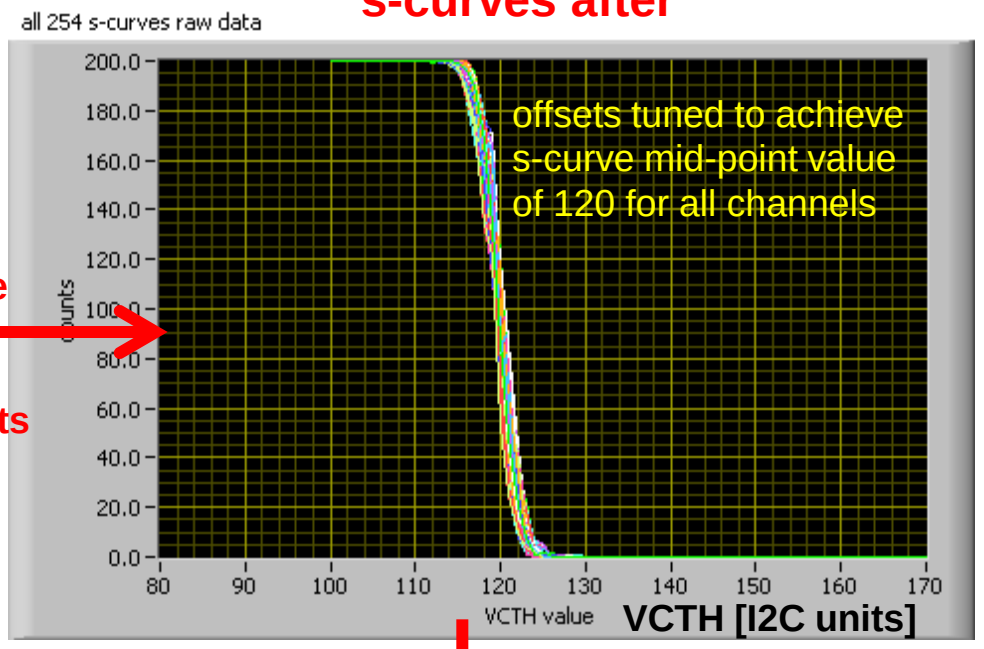
offsets tuning - now sweeping VCTH (VPLUS fixed at 111_d)

s-curves before tuning

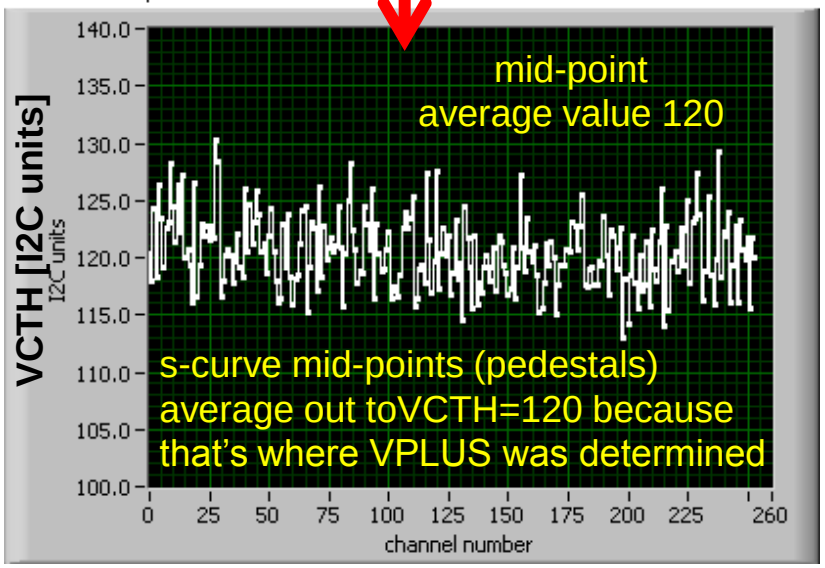


tune the offsets

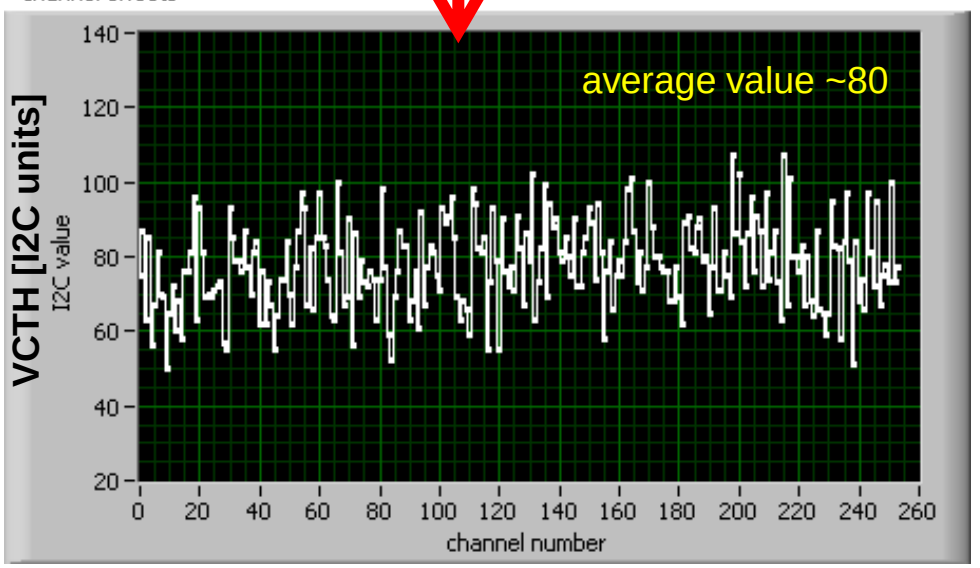
s-curves after



mid-points vs. channel no.



offset vs. channel no.



final offsets tuning strategy proposal

set all channel offsets to 80_d

set VCTH to 120_d

sweep VPLUS to find value corresponding to average channel s-curve midpoint

that is final choice for VPLUS

now tune channel offsets in the normal way to achieve midpoint of 120_d

tune pedestals (not using test pulse) in groups of 32

individual channel offset values will vary channel-to-channel (but average to ~80)