Gain History

Note: these gain numbers are calculated using "flat field" images, where each pixel has a small number of photons (1-3). They may not be valid for high photon fluxes.

CsPad per-pixel gains from flat field measurement

Date	Detector	Gain mode	Photon source	Occupancy (photons /pixel)	Per-pixel histogram fit HDF5	Experiment /Run	Deployed gain file	Comments
2016-02-16	DsaCsPad	high	Iron K-alpha (6.4 keV)	0.015	/reg/g/psdm/detector/alignment/cspad/ calib-cxi-camera2-2016-02-05/calib/ CsPad::CalibV1/CxiDs2.0:Cspad.0/pixel_gain/ run16_DsaCsPad_flat_orig.h5	cxi01516/16	cxi01516-r0016-2016-02-18-FeKalpha.data	Success rate was low ~0.75%. Filled in the rest with average gain from each asic.
2016-04-13	DsaCsPad	high	Cu K-alpha (8.0keV)	0.043	/reg/g/psdm/detector/alignment/cspad/ calib-cxi-camera2-2016-02-05/calib/ CsPad::CalibV1/CxiDs2.0:Cspad.0/pixel_gain/ run6_DsaCsPad_flatIso.h5	cxi00516/6	run6_DsaCsPad_flatIso.data	Success rate was high ~0.99%, but there were some scattering in the centre, so filled that in with the average gain from each asic. Experiment: cxi00516 run6
2018-04	DsaCsPad		Cu			cxilr6716	gain_prelim.npy	From Derek Mendez: Its not perfect but seems to improve the data (pretty important for the cor relations experiment). Interestingly in this camera there is also a lone quad that is off. The initial goal was to measure an intensity-dependent gain correction (hence the different transmissions), but for now I just use the average gain cal from these data. One such average-gain map is stored as a 3D numpy array (same shape array returned by det.raw(event)), this is in cxilt1417 /scratch/gain/gain_prelim.npy* . the correction would be >> gain = np.load("gain_prelim.npy") >> assert (np.all(gain!=0)) >> data = det.calib(None) >> data /= gain

Note: CsPad has high and low gain modes. A factor of 7 is used to recalibrate the high gain to low gain, e.g. 28 ADU at high gain results in 4 ADU at low gain.

Calculation note: At 8keV, we expect around 30 ADUs per photon in high gain mode. At 6.4keV, we are seeing around 23 ADU per photon. currentADU /currentEnergy*8keV=23/6.4*8=28.75 ADU per photon which is close to 30 ADUs.

K-alpha table for Iron (6.4keV):

http://www.yourperiodictable.com/iron.php

K-alpha table for Copper (8.0keV):

http://www.yourperiodictable.com/copper.php

Pnccd per-pixel gains from flat field measurement

Date	Detector	Gain mode	Photon source	Occupancy (photons /pixel)	Per-pixel histogram fit HDF5	Experiment /Run	Deployed gain file	Comments
2016-04-15	pnccdFront	6	Si K-alpha (1.74keV)	0.006	/reg/g/psdm/detector/alignment/pnccd/ amo06516-pnccd-2016-04-14/calib/ PNCCD::CalibV1/Camp.0:pnCCD.0/pixel_gain/ run10_pnccdFront_flatIso_clean.h5	amo06516/10	run10_pnccdFront_flatIso_clean.data	Success rate was high ~0.96%. Filled in the rest with average gain from each 128 pixel strips.
2016-04-15	pnccdFront	5	Si K-alpha (1.74keV)	0.004	/reg/g/psdm/detector/alignment/pnccd/ amo06516-pnccd-2016-04-14/calib/ PNCCD::CalibV1/Camp.0:pnCCD.0/pixel_gain/ run15_pnccdFront_flatIso_clean.h5	amo06516/15	run15_pnccdFront_flatIso_clean.data	Success rate was high ~0.96%. Filled in the rest with average gain from each 128 pixel strips.
2016-04-15	pnccdFront	4	Si K-alpha (1.74keV)	0.003	/reg/g/psdm/detector/alignment/pnccd/ amo06516-pnccd-2016-04-14/calib/ PNCCD::CalibV1/Camp.0:pnCCD.0/pixel_gain/ run19_pnccdFront_flatIso_clean.h5	amo06516/19	run19_pnccdFront_flatIso_clean.data	Success rate was high ~0.95%. Filled in the rest with average gain from each 128 pixel strips.

Note: Pnccd has numbered gain modes.

Kaz says that PNCCD gives out ~1250ADU per 1keV photon in highest gain range.

K-alpha table for Silicon (1.74keV):

http://www.yourperiodictable.com/silicon.php

EPIX per-pixel gains

NOTE: **In 2017 we have started using the epix id numbers with the new global calibration store. So this epix gain section is now deprecated. The global calibration store is at /reg/d/psdm/detector/calib/ and can be examined with the "dcs" command in psana.**

Date	Photon Source	Carrier ID 0 /1	Digital Card ID 0/1	Analog Card ID 0/1	Experiment /Run	Gain File	Comment
2016-04-14	Cu K-alpha (8 keV)	996663297 /3791650826	1232098304 /953206283	2655595777 /520093719	xcs01116/95	gainPixelCorr_passZero_r95_c1.txt	From Philip Hart
2016-06-30	Cu K-alpha	996477441 /2264924170	1794137088 /403490827	2397900801 /4076863512	xcs06016/37	gainPixelCorr_passZero_r37_c1.txt	From Philip Hart
2016-06-30	Cu K-alpha	996579585 /553648138	1232098304 /1221641739	2650251521 /3976200215	xcs06016/37	gainPixelCorr_passZero_r37_c2.txt	From Philip Hart
2016-06-30	Cu K-alpha	996663297 /3791650826	1232098304 /953206283	2655595777 /520093719	xcs06016/37	gainPixelCorr_passZero_r3335_c3.txt	From Philip Hart
2016-07-01	Ni fluorescence	998779393 /117440522	1794137088 /403490827	2669921025 /100663319	xcs06016/52	gainPixelCorr_passZero_r52_c2.txt	From Philip Hart
2016-07-01	Ni fluorescence	996513537 /2080374794	1794135040 /940361739	2398406657 /419430424	xcs06016/52	gainPixelCorr_passZero_r52_c3.txt	From Philip Hart
2016-07-01	Cu K-alpha	996432897 /3590324234	1232100352 /1154532875	2654088449 /33554455	xcs06016/37	gainPixelCorr_passZero_r37_c0.txt	From Philip Hart
2016-10-21		As above			xcsm9816/2-27,45-49	gainPixelCorr_passZeroL_r2049_c2.txt	From Philip Hart
2017-06-06	Cu K-alpha	997010433 /3170893834	1794137088 /403490827	2403244545 /1224736792	mfx01316/6,7	gain3PixelCorr_passZero_r6007_c0.txt	From Philip Hart, fit statistics might be marginal so consider using the alternative non-fit results below
		As above				gainPixelCorr_passZero_r6007_c0.txt	From Philip Hart, uses mode instead of gaussian fit because of marginal statistics
2017-09-22	Fe55	0996513537 /2080374794	1794135040 /0940361739	2398406657 /0419430424	detdaq17 9-15	gainPixelCorr_passZeroL_r10121315_c0.npy	From Philiph Hart

The best way to access epix10k ID numbers is with this command:

```
epix10ka_id exp=mfxc00318:run=13 epix10k2M
```

An uglier approach is to use the following script:

```
from psana import *
import sys
ds = DataSource(sys.argv[1])
evt = ds.events().next()
cs = ds.env().configStore()
cfg = cs.get(Epix.Config100aV2, Source(sys.argv[2]))
print str(cfg.version()) # not ideal for epix10ka2M (and quads): currently the config DDL version (cfg.
Version).  ideally would be the firmware version (but didn't exist for early versions)
print str(cfg.carrierId0())+'/'+str(cfg.carrierId1())
print str(cfg.digitalCardId0())+'/'+str(cfg.digitalCardId1())
print str(cfg.analogCardId0())+'/'+str(cfg.analogCardId1())
#a0 = cfg.asics(0)
#a0.chipID()
```

Which can be run like this:

```
python epixSerialNumber.py exp=mecl8216:run=73 epix100a
```