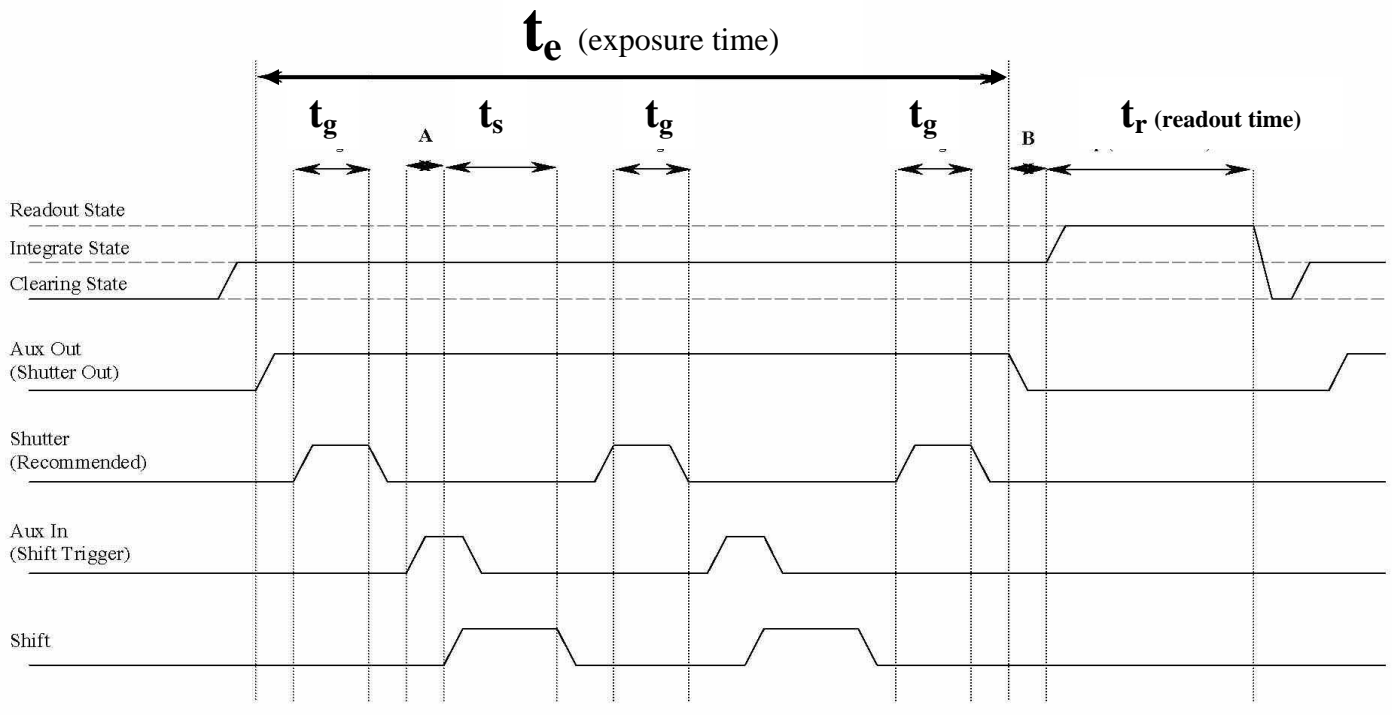


Date: November 20, 2006
 Re: Mar165's Frameshift at APS

Below is Mar-USA's "documentation" for their frameshift option on the Mar165, which is installed on the Detector Pool's mar165A (S/N: 62). (Note: the version of the marccd_server_socket program changed since Mark Rivers wrote his EPICS ccdApp. So, you have to be careful about this. Check the maintenance log for the mar165A for a description about how to fix this problem.)



t_e = Exposure time as set in marccd dialog (single frame or dataset dialog)
 t_r = Readout time characteristic of CCD, binning, speed setting, PSD
 t_g = Gate time – exposure time of individual sub-exposures – determined by user's gating of shutter
 t_s = Shift time = $N * t_l$
 N = Number of line pairs (N is set in software, height of each corrected subimage is $2*N$ lines)
 t_l = Shift time per line pair = $a + b * PSD$, $a = 32\mu s$ $b = 1.6\mu s$, Normal value is $\sim 440\mu s$, minimum is $\sim 34\mu s$
 PSD = Parallel Shift Delay = Software selectable from 1 to 4095, Normal is 250, full dynamic range guaranteed only at 250
 A = Jitter between Aux In and start of shift, approximately 1ms
 B = Shutter close delay – software selectable – normally 1 to 40ms

t_e = Exposure time as set in marccd dialog (single frame or dataset dialog)

t_r = Readout time characteristic of CCD, binning, speed setting, PSD

t_g = Gate time – exposure time of individual sub-exposures – determined by user’s gating of shutter

t_s = Shift time = **$N * t_l$**

N = Number of line pairs (**N** is set in software, height of each corrected subimage is $2*N$ lines)

t_l = Shift time per line pair = $a + b * PSD$, $a = 32 \mu s$ $b = 1.6 \mu s$, Normal value is $\sim 440 \mu s$, minimum is $\sim 34 \mu s$

PSD = Parallel Shift Delay = Software selectable from 1 to 4095, Normal is 250, full dynamic range guaranteed only at 250

A = Jitter between Aux In and start of shift, approximately 1ms

B = Shutter close delay – software selectable – normally 1 to 40ms

The idea with the frameshift (aka “kinetics” or “streak” mode) option is that you use a portion the CCD chip to buffer a small portion which is exposed to x-rays. So, you can take a few exposures quickly before having to readout the CCD. So, you will need to mask out the portion of the CCD chip which is used to buffer the images. The CCD chip is a 4k x 4k chip, but can’t be readout in that resolution. 2k x 2k is the highest resolution (2x2 binning) (about 80 micron pixels at the phosphor). The chip has 4 output amplifiers at the each of the corners. In frameshift mode, the user specifies the number of lines that will be used in the exposure and mask out the remainder of the camera. In the mar165s normal position, the stripe is horizontal. The marccd software will correct the image (background, flat field, and distortion) appropriately for the frameshift mode. In this mode, the exposures actually occur on the same part of the CCD for all exposures. Note, that is you let marccd to the correction, it will reorder the images so that in the tiff file, the stripes will ordered from top to bottom in the order that they were taken. If you don’t apply the correct, you will see get exactly what the raw readout image is. Half will be on top and other half on the bottom. This is because the CCD chip is readout of 4 readout amplifiers at the corners.

The timing diagram above describes how to control the CCD using external trigger pulse. Via software, you need to set the number of lines to shift (e.g., thru the remote control interface: `set_frameshift,100`). With this command sent, the CCD will shift charge vertically by this 100 rows when a TTL pulse is sent to “Aux In”. (The exact specifications of the TTL pulse are not specified or at least known to Michael Blum. So, use something wide enough to work, but not too long and something close to 5V. The first implementation of the frameshift was done at 33-id with Jon Tischler.) On the controller there is an “AUX control” connector which a 4-pin LEMO connector. This has connection for “trigger in” which is “Aux In” for shifting charge. The “trigger out” is used control the x-ray shutter. This goes positive when the detector goes into the

“integrating” state, which is determined by the detector, and then negative at the end of the exposure before the readout. So, the detector cannot really put into a “slave” mode.

There are several parameters that can be adjusted via software. In order to access these parameter, you need to start marccd with a special proprietary flag:

marccd -P

If you go to Configure → Detector, you will get the window to “reboot” the controller And start the cooler. Also, you will see various parameters you can vary. These include:

Parallel Shift Delay (PSD): This allows you to change the shift time per line pair. Mar-USA suggests using 250. They say you would lose dynamic range if smaller, but I think they mean charge transfer efficiency will suffer.

There is also a Shutter Close Delay, if you would like to use it. It also appears that only 2x2 binning is allowed for frameshift mode.

At APS, we typically use Mark River’s EPICS ccdApp interface (distributed as part of BCDA’s synApps). Mar-USA provides a marccd_socket_server with a set of instructions to control the marccd. In the marccd GUI, you need to start the remote control mode (Acquire → Remote Control). At the core of ccdApp, is an EPICS sequence program (marCCD.st) which sends various commands to the marccd_socket_server. In the “Setup” section of the ccdApp medm screen, there is an asynRecord window where you can send commands to the marccd_socket_server for debugging purposes. Note, that there are 2 versions of marccd_socket_server (0 and 1). ccdApp uses version 0, but the The default from Mar-USA is 1. So, if you upgrade software, beware of this. Things won’t work. There is a way to change marccd_socket_server to version 0. Check the maintenance log file for the mar165A for instructions.

For example, a background image should be acquired before a regular exposure is taken. The commands to the marccd_socket_server would look like:

Here is a sequence that will make a dezingered bias frame (recommended method):

- [CLOSE SHUTTER] *(make sure shutter is closed)*
- start *(start integration)*
- readout , 2 *(read and copy to Scratch buffer)*
- start
- readout , 1 *(read and copy to Background buffer)*
- dezing , 1 *(dezing from Background and Scratch data, put image in Background buffer)*

The background doesn't have to be retaken for every data image taken, but generally should be retaken at the start of every new data set, or once every half hour, whichever is sooner (depending on the thermal stability of the hutch). For the MarCCD detector, if a mismatch in the level of the 4 quadrants of data frames is noticed, the bias is probably drifting and should be recollected (and maybe should be set to be collected more often).

To collect a data image:

- `start` (start integration)
- `[OPEN SHUTTER]`
- `[WAIT DESIRED TIME]`
- `[CLOSE SHUTTER]`
- `readout, 0` (read data into raw frame buffer)
- `correct` (apply correction; data sent to "corrected" buffer)
- `writefile, FILENAME, 1` (write data from corrected frame buffer to file)

The new frameshift software has 2 new commands.

`set_frameshift, nlines` (sets the number of line pairs to shift)
`get_frameshift` (returns the value of nlines) (May not work??)

At the moment, I have not modified `ccdApp`'s `medm` screen to include an option to change into the frameshift mode, but you can do this by hand by sending the `set_frameshift, nlines` command from the `asynRecord` screen. (Or do in the `marccd` GUI when you "Reboot" the controller. It just needs to set once.

Remember, to take background images if you every time you make changes (e.g., change the binning). Better safe than sorry. You would see error messages in the terminal window where you started the `marccd` GUI.

Now, let's describe how frameshift was implemented at 33-id in June 2006 by Jon Tischler. Once you have set the `marccd` into the frameshift mode, you need synchronize shift of rows with the shutter and the readout of `marccd`. The typical usage at 33-id was to perform 0.5 second exposures for 20 stripes. This consists of a total of 10 seconds with the shutter open. The "Aux Trigger Out" was connected to the x-ray shutter. This shutter stays open for the entire exposure. An external function generator is connected to "Aux Trigger In". This is programmed independently from the `marccd`, but must be synchronized. For 20 stripes with 0.5 second exposures, this means that 19 pulses are sent every 0.5 seconds. The function generator is triggered using the "Aux Trigger Out". Assuming that the frameshift parameter has been set and background images have been taken, the frameshift exposures can begin via the normal `ccdApp` "Start" button (or via spec if you know how to do that). The exposure time should be set to appropriate value given the number of shifts given. If not, the readout will happen at the wrong time and all the stripes might not have the same exposure times. Once `marccd` is ready to expose, it will change its internal state to "Acquire" (i.e., state = 1). This will send "Aux Trigger

Out” high, which will open the x-ray shutter and trigger the function generator to shift at appropriate times. If everything is synchronized properly, the ccdApp will send out the readout command at the right time and also do the corrections and write the file to disk. In this implementation, we did not close the shutter during the shifting of lines which results in some streaking. However, since the exposure time is much longer than the shifting time, the streaking is not a big effect.