NSLS-II CSX Beamline Docs Documentation

Release 0.1

Stuart B. Wilkins

November 24, 2015

CONTENTS

Contents:

CHAPTER

ONE

CSX-1 (23-ID-1) BEAMLINE DOCCUMENTATION

Contents:

1.1 Fast CCD Detector

1.1.1 Introduction

The FastCCD installed in the endstation at CSX-1 is of the LBNL Fast CCD design. The sensor contains 1920 x 960 pixels of 30 µm x 30 µm and is arranged into two halves of 960 rows by 960 columns with the columns parallel to the long CCD axis. There is one output for each 10 columns (a "super column") which results in 192 individual outputs and analogue to digital converters (ADC). The CCD camera can either be used in a traditional CCD with an x-ray shutter exposing the full chip, or in a framestore (frame transfer) mode by covering two quarters of the CCD with a light (x-ray) block effectively exposing half the chip along the column direction.

The analogue CCD signal is digitized by a custom designed fCRIC. Each fCRIC has 16 analogue inputs and digitizes with 13 bit precision and had 16 bit dynamic range. This is accomplished by having 3 gain ranges of 8x, 4x and 1x with an auto gain feature. In order to allow negative charge injection. The ADC is biased at a value of approximately 4096 (0x1000 in hex) with the exact value dependent on the ADC channel. The gain settings are stored in the two most significant bits of each ADC reading. The schematic of a single fCRIC chanel is shown in the *LBNL fCRIC Circuit Diagram*.

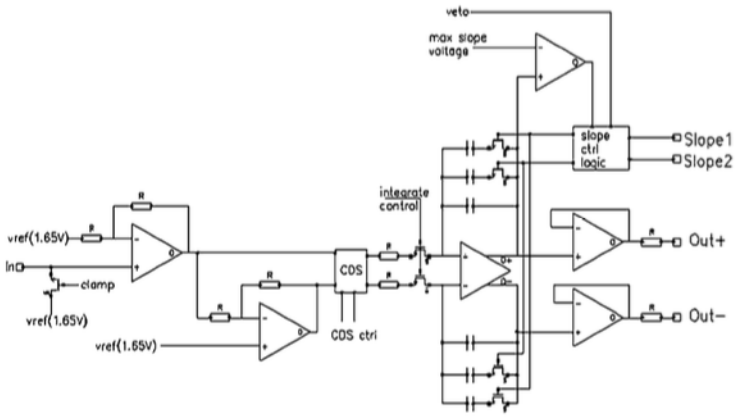

Fig. 1.1: LBNL fCRIC Circuit Diagram

The specifications of the CCD are summarized below:

- Pixel Size: 30 µm x 30 µm
- Active Area: 1920 pixels (column) x 960 pixels (row)
- 192 super columns = 192 outputs (480 rows x 10 columns)
- Back illuminated
- 250 µm - 350 µm thickness
- Full well : ~900k e^- per pixel
- Sensitivity : 6 e^- / ADU for 8x gain (max gain)
- Pixel readout time: 500 µs
- Digitization time: 2 µs at 120 Hz
- 100 Hz maximum data collection

1.1.2 Data Format

In treating the raw CCD data from the FastCCD there are a few important considerations related to the multi-gain behaviour of the fCRIC amplifier and digitizer. The raw 16 bit values that are recorded in the data file follow the *16 Bit fCRIC Data Format* shown below with the two gain bits following the *fCRIC Gain Setting*.

Table 1.1: 16 Bit fCRIC Data Format

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
G1	G0	ERR	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00

Table 1.2: fCRIC Gain Setting

G1	G0	Gain	Pre-factor
0	0	x8	x1
1	0	x2	x4
1	1	x1	x8

Here the two most significant bits record the gain setting for the encoded value. The least significant 13 bits hold the measured analogue value. The actual value is therefore related to the measured value by:

$$A_{\mathrm{corr}} = G(A_{\mathrm{meas}} - O)$$

where A_{corr} is the corrected intensity, A_{meas} is the measured value by the ADC, G is the gain of the ADC and O is the bias offset.

1.1.3 Dark Image Subtraction

Due to the multi gain nature of the fCRIC it is therefore necessary to take 3 dark images at different gain settings to obtain the different ADC offsets under these modes. As the lower gain settings are not subject to considerable contribution due to dark current it is usually justifiable to measure only the highest gain dark image repeatedly. Given 3 dark images for the different gain settings the images the following python pseudo code can be used to correct for dark current and gain:

```
import numpy as np

def subtract_background(image, dark_image, gain = [1, 4, 8]):
    gain_mask_8 = (image & 0xC000) == 0xC000
    gain_mask_4 = (image & 0xC000) == 0x8000
    gain_mask_1 = (image & 0xC000) == 0x0000
```

```
    cor_image = image.astype(np.float16)
    cor_image -= gain_mask_8 * dark_image[2]
    cor_image -= gain_mask_4 * dark_image[1]
    cor_image -= gain_mask_1 * dark_image[0]

    gain_image = (gain_mask_8 * gain[2]) + (gain_mask_4 * gain[1]) + (gain_mask_1 * gain[0])

    return (cor_image * gain_image), gain_image
```

1.1.4 Useful Links

- LBNL Fast CCD Site
- csxtools python analysis routines
- libcin low level c driver
- areaDetector Driver

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

THREE

HELP!! THE %$^$#@% JUST CRASHED

3.1 Managing IOCs

Soft IOCs are managed through the `manage-iocs` script. To obtain a list of softiocs running on a NSLS-II computer use the command `manage-iocs report` an example is shown below for `xf23id1-ioc3`:

```
[swilkins@xf23id1-ioc3 ~]$ manage-iocs report
nBASE            | IOC             | USER            |  PORT | EXEC
/epics/iocs      | apcupsd         | root            |  5000 | /epics/iocs/apcupsd/st.cmd
/epics/iocs      | cam-diag1       | softioc         |  4202 | /epics/iocs/cam-diag1/st.cmd
/epics/iocs      | cam-diag6       | softioc         |  4300 | /epics/iocs/cam-diag6/st.cmd
/epics/iocs      | cam-dif1        | softioc         |  4204 | /epics/iocs/cam-dif1/st.cmd
/epics/iocs      | cam-dif2        | softioc         |  4205 | /epics/iocs/cam-dif2/st.cmd
/epics/iocs      | cam-dif3        | softioc         |  4206 | /epics/iocs/cam-dif3/st.cmd
/epics/iocs      | cam-dif-beam    | softioc         |  4201 | /epics/iocs/cam-dif-beam/st.cmd
/epics/iocs      | ct-eps          | softioc         |  4002 | /epics/iocs/ct-eps/st.cmd
/epics/iocs      | es-dg645        | softioc         |  5013 | /epics/iocs/es-dg645/st.cmd
/epics/iocs      | es-K2611        | softioc         |  4302 | /epics/iocs/es-K2611/st.cmd
/epics/iocs      | es-tctrl1       | softioc         |  5010 | /epics/iocs/es-tctrl1/st.cmd
/epics/iocs      | es-vortex       | softioc         |  4301 | /epics/iocs/es-vortex/st.cmd
/epics/iocs      | mc11            | softioc         |  5001 | /epics/iocs/mc11/st.cmd
/epics/iocs      | mc12            | softioc         |  5002 | /epics/iocs/mc12/st.cmd
/epics/iocs      | mc13            | softioc         |  5003 | /epics/iocs/mc13/st.cmd
/epics/iocs      | omegaM4061      | softioc         |  5012 | /epics/iocs/omegaM4061/st.cmd
/epics/iocs      | simdetector     | softioc         |  4203 | /epics/iocs/simdetector/st.cmd
/epics/iocs      | simmotor        | softioc         |  8001 | /epics/iocs/simmotor/st.cmd
/epics/iocs      | timestamp       | softioc         |  6001 | /epics/iocs/timestamp/st.cmd
/epics/iocs      | va-bakeout-01   | softioc         |  4001 | /epics/iocs/va-bakeout-01/st.cmd
/epics/iocs      | zebra           | softioc         |  5011 | /epics/iocs/zebra/st.cmd
```

To connect to the IOC console, telnet to localhost at the port that is shown in the table. For example to connect to the `mc12` console issue the command:

```
[swilkins@xf23id1-ioc3 ~]$ telnet localhost 5002
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
@@@ Welcome to procServ (procServ Process Server 2.6.0)
@@@ Use ^X to kill the child, auto restart is ON, use ^T to toggle auto restart
@@@ procServ server PID: 10584
@@@ Server startup directory: /epics/iocs/mc12
@@@ Child startup directory: /epics/iocs/mc12
@@@ Child "mc12" started as: /epics/iocs/mc12/st.cmd
@@@ Child "mc12" PID: 28044
```

```
@@@ procServ server started at: Tue Oct 20 17:35:25 2015
@@@ Child "mc12" started at: Fri Nov 13 12:49:49 2015
@@@ 0 user(s) and 0 logger(s) connected (plus you)
```

In order to reboot the IOC, type `[CTRL] + X`. To leave the console type `[CTRL] + ]` and type `close` at the `telnet>` prompt

To start all IOCs configured on the system issue the command `sudo manage-iocs startall` and if needed to stop all IOCs issue the command `sudo manage-iocs stopall`

3.2 OLog Glassfish Server

To reboot the glassfish server on `xf23id-ca.cs.nsls2.local` execute:

```
swilkins@xf23id-ca:~$sudo su - glassfish
glassfish@xf23id-ca:~$cd glassfish3/bin/
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin stop-domain domain1
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin stop-domain domain2
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin start-domain domain1
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin start-domain domain2
```

CHAPTER

FOUR

UNIX PRIMER AND ACCOUNT SETUP

4.1 File Permissions

In a unix system every file (and directory) has an owner, an associated group, and a set of permission flags that specify separate read, write, and execute permissions for the *user* (owner), *group*, and *other*. *other* is also sometimes known as *world* permissions, and applies to all users who can login to the system.

The permission flags can be seen by the `ls -l` command:

```
[swilkins@xf23id1-srv1 ~/Presentations]$ ls -l
total 24M
-rw-rw---- 1 swilkins csx  22M Mar  5  2014 csx.pdf
-rw-rw---- 1 swilkins csx 1.2M Mar  3  2014 csx.ppt
drwxrwx--- 2 swilkins csx    6 Nov 24 06:15 other
```

The first field is a set of 10 permission flags. The first flag is the *file type*. For regular files it is not set (-). Directories are indicated by *d* and links by *l*. The remaining 9 flags are 3 groups of 3 for read (r) / write (w) / execute (x) for the user, group and other. In the example above the files are read/write permitted for both the user (swilkins) and any members of the csx group. Any other users cannot read or write the files. Note that the directory *other* has the execute bit set for only the user and group to restrict listing to only the user and members of the csx group.

To set file permissions, UNIX uses the *octal* representation of file permissions, with 4 digits representing the special, user, group and other. These permissions are outlined in the table *octal file permissions*

Table 4.1: Octal file permissions for chmod and umask

Octal digit	`chmod` Permissions	`umask` Permissions
0	No permissions	Any permission may be set
1	Execute permission only	Setting of execute permission is prohibited
2	Write permission only	Setting of write permission is prohibited
3	Write and execute permissions	Setting of write and execute permission is prohibited
4	Read permission only	Setting of read permission is prohibited
5	Read and execute permissions	Setting of read and execute permission is prohibited
6	Read and write permissions	Setting of read and write permission is prohibited
7	Read, write and execute permissions	All permissions are prohibited from being set

The file permissions can be changed using the `chmod` command with the permissions specified in an octal format and can be applied *recursively* using the `-R` option. For example:

```
[swilkins@xf23id1-srv1 ~/Presentations]$ ls -l
total 24M
-rw-rw---- 1 swilkins csx  22M Mar  5  2014 csx.pdf
-rw-rw---- 1 swilkins csx 1.2M Mar  3  2014 csx.ppt
drwxrwx--- 2 swilkins csx    6 Nov 24 06:15 other
```

```
[swilkins@xf23id1-srv1 ~/Presentations]$ chmod 700 other
[swilkins@xf23id1-srv1 ~/Presentations]$ ls -l
total 24M
-rw-rw---- 1 swilkins csx  22M Mar  5  2014 csx.pdf
-rw-rw---- 1 swilkins csx 1.2M Mar  3  2014 csx.ppt
drwx------ 2 swilkins csx    6 Nov 24 06:15 other
[swilkins@xf23id1-srv1 ~/Presentations]$
```

UNIX uses the **umask** to define how (new or default) files permissions are set. This can be set on a per-user basis using the `umask` command to set the default (allowed) permissions. The `umask` command uses the *octal representation* the same as `chmod` to define permissions and should have the *restricted* permissions set - i.e. the inverse of what you want the files to be.

```
$ umask 0227      # allow user rwx group r-x and other r-x      (-rwxr-xr-x)
$ umask 0027      # allow user rwx group r-x and restrict other (-rwxr-x---)
$ umask 0007      # allow group and user rwx and restrict other (-rwxrwx---)
$ umask 0077      # allow only user rwx                         (-rwx------)
```

The following example illustrates the use of the `umask` command:

```
[swilkins@xf23id1-srv1 ~/Example]$ umask
0007
[swilkins@xf23id1-srv1 ~/Example]$ touch hello_world
[swilkins@xf23id1-srv1 ~/Example]$ ls -l
total 0
-rw-rw---- 1 swilkins csx 0 Nov 24 06:55 hello_world
[swilkins@xf23id1-srv1 ~/Example]$ umask 0027
[swilkins@xf23id1-srv1 ~/Example]$ touch goodbye_world
[swilkins@xf23id1-srv1 ~/Example]$ ls -l
total 0
-rw-r----- 1 swilkins csx 0 Nov 24 06:55 goodbye_world
-rw-rw---- 1 swilkins csx 0 Nov 24 06:55 hello_world
[swilkins@xf23id1-srv1 ~/Example]$ umask 0023
[swilkins@xf23id1-srv1 ~/Example]$ touch hello_world_again
[swilkins@xf23id1-srv1 ~/Example]$ ls -l
total 0
-rw-r----- 1 swilkins csx 0 Nov 24 06:55 goodbye_world
-rw-rw---- 1 swilkins csx 0 Nov 24 06:55 hello_world
-rw-r--r-- 1 swilkins csx 0 Nov 24 06:58 hello_world_again
```

To set the *umask* for every session the command should be added to your `.bashrc` file. CSX Team members should add the following to their `.bashrc`:

```
umask 0027
```

4.2 Bash Shell Setup

The BASH shell is initialized by two main files `.bashrc` and `.bash_profile`. These are executed by the shell differently depending on how the shell is executed. This is often a source of confusion. These files are summarized below:

- `.bashrc` : This is run by *interactive* shells. These are shells that are connected to a terminal (or pseudo-terminal) such as a *xterm* running under a windowing system.
- `.bash_profile` : This is run by *login* shells. These are shells that are started when you login from another host or you login from the text terminal on a local machine.

BASH is also different from other shells in that `.bashrc` and `.bash_profile` are *mutually exclusive* (i.e. only one is run on the shell startup). To get round this problem, most people place teh following in the `.bash_profile` so that the shell is initialized the same way for both *interactive* and *login* shells:

```
if [ -f ~/.bashrc ]; then
    source ~/.bashrc
fi
```

CHAPTER

FIVE

DOWNLOADS

Download the CSX Documentation as a PDF

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search