Archon Readout Notes for ZTF

D. Hale
Caltech Optical Observatories
M/S 11-17, 1200 E California Blvd., Pasadena, CA 91125

October 30, 2014

Contents

List of Figures

List of Tables

1 Abstract

This document describes the basic architecture of STA's Archon CCD controller and how it might be utilized by ZTF. It is anticipated that this document may form the basis for a future design document.

2 Archon

2.1 Overview

Archon is a high performance modular CCD controller developed by Semiconductor Technology Associates, Inc (STA). An Archon system receives configuration information from and sends status and image data to a host computer via a gigabit Ethernet connection. Power is applied to Archon through a circular connector carrying the DC voltages necessary for a particular system. The CCD(s) to be operated is(are) connected to Archon through a custom interface board, built to route signals from the CCD cabling to the internal Archon module connectors.

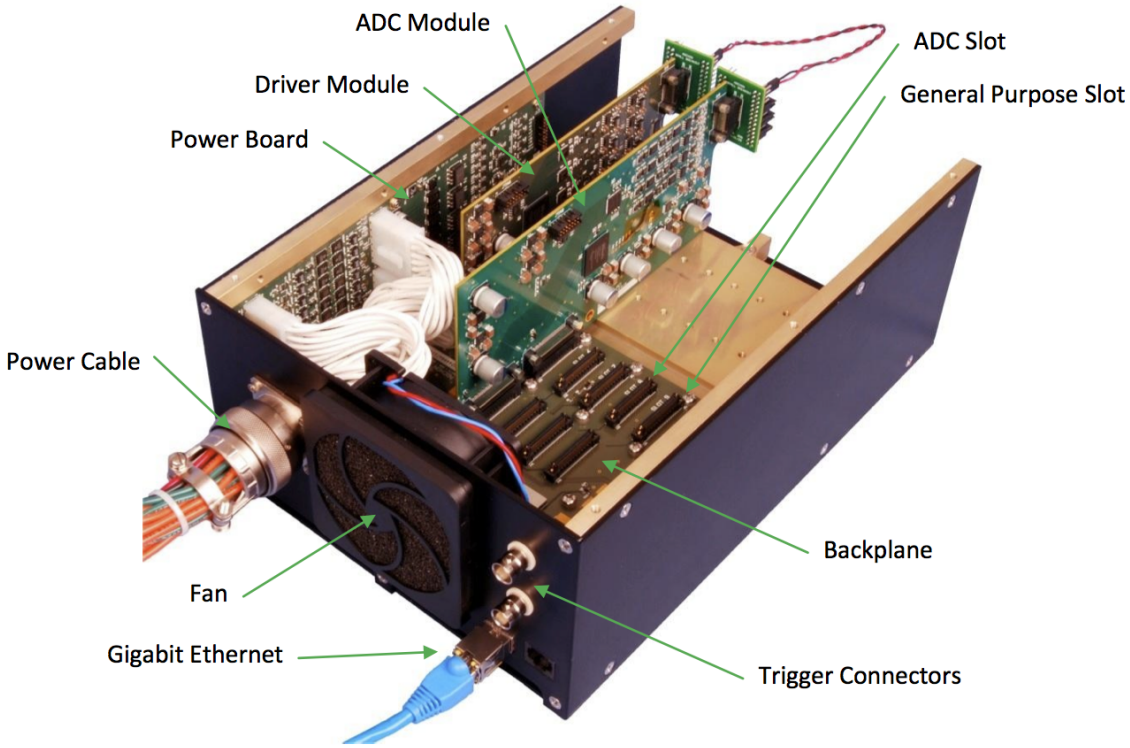

Figure 1: Example Archon controller

2.2 Terminology

The following describes some of the terminology used by STA which might be different from that seen elsewhere:

Module
Card in a slot. There are different types of cards (modules), as follows:

Driver Module
Clock board. Each clock board has 8 channels.

ADC Module
Video board. Each video board has 4 channels (called "taps"). There can be up to four ADC (video) modules in a controller. These video boards must be installed in specific slots. The board identity (i.e., the channel numbers) are determined by the slot in which it is installed.

HV Bias Module
DC bias generator board. 30 programmable biases, 0 to +31v

LV Bias Module
DC bias generator board. 30 programmable biases, −14 to +14v

Tap
channel

Configuration
Technically a Windows INI-style configuration file of keyword and value pairs. The configuration file sets states and variables used to control the the backplane microprocessor and the installed modules. Loosely speaking, one might refer to this as the clock waveforms, although it controls everything in the Archon controller and all installed modules, such as clock timing and levels, sample and hold periods, deinterlacing modes, etc.

2.3 Backplane

The Archon backplane is responsible for communication with the host system and the installed modules. The X12 variant can support 12 modules, including 4 ADC Modules (i.e., video boards). Processing is done by a 32 bit softcore microprocessor embedded in the backplane FPGA. The microprocessor has 2GB of RAM, 512MB of which is reserved for the processor and 1.5GB for frame buffers. 16MB of flash memory stores firmware and controller configuration data.

3 Frame Buffers

The Archon controller allocates 1.5GB of RAM to frame buffer memory. This memory can be configured as three 512MB buffers or two 768MB buffers. ZTF will require the latter. To enable 2×768MB buffers, define the key `BIGBUF=1` in the configuration file.

The backplane microprocessor keeps track of the frame buffers and their state, i.e., which is available for reading or writing and which is next in line to be used. It also maintains an incremental frame number (xxxx bits). At power-up the frame number is initialized to 0, but the first frame read will be called frame 1. When the frame number reaches xxxx it yyyy.

4 Deinterlacing

4.1 Configuration

Deinterlacing is performed by the Archon controller; the pattern is defined in the configuration file by the keys `TAPLINES`, `TAPLINEn`, `PIXELCOUNT`, `LINECOUNT`, `FRAMEMODE` and `SAMPLEMODE`. A description of these keys is as follows. Example usages will be given below. (Recall that a "tap" is a channel in the ADC i.e., video board. A CCD with four outputs will use four taps.)

TAPLINES
the total number of tap configuration lines defined in the configuration file, 0–63.

TAPLINEn
the actual configuration file keyword, n=0–63. This key has the format,

`TAPLINEn=''tap,gain,offset''`
where "`tap`" is a string of the form:

`ADnd`
where n is the ADC (video) channel number 1–16 and d is 'L' or 'R'. The video channel is defined by the location of the ADC Module in the backplane. Use n=1 for the first channel of the first ADC module installed in slot 5, n=2 for the second channel, etc. up to 4. The four channels of the second ADC module (installed in slot 6) would use n=5–8, and so on up to n=16 for the 4th channel of the 4th ADC module (installed in backplane slot 8).

The readout direction (d) is "L" if the first pixel values should be written at the left edge of a tap's portion of the frame buffer, or "R" to start at the right edge.

"`gain`" is a floating point gain value.

"`offset`" is an integer to add to the pixel values.

PIXELCOUNT
the number of pixels (columns) per tap (channel), limit? TBC.

LINECOUNT
the number of lines (rows) per tap (channel), 1–65535.

FRAMEMODE
defines the frame deinterlacing mode as follows:

`FRAMEMODE=0`
Top; the first pixel is written to the top of the frame.

`FRAMEMODE=1`
Bottom; the first pixel is written to the bottom of the frame.

`FRAMEMODE=2`
Split; the first half of the taps are written to the top and the second half are written to the bottom of the frame.

SAMPLEMODE
0 is 16 bit and 1 is 32 bit (ZTF would use `SAMPLEMODE=1`).

4.2 Examples

The test pattern used in the following examples was selected to help identify the location of pixels in the frame and illustrate Archon's deinterlacing scheme. The image is 3072 × 3080. The first transmitted 300 rows are a fixed value. After that a left-to-right gradient is created by increasing the value every 384 columns. Thus, eight bands are seen left-right (after the initial fixed-value rows). The first band is the lowest value (represented by the darkest shade) and increases in value (shade brightness increasing) to the right.

By convention, FITS images are displayed with the origin at the lower left, with row number increasing upward. For this document the row orientation has been reversed, with row 0 being displayed at the top. This has been done so that the displayed images are consistent with the terminology used by Archon (i.e., "Top" deinterlacing shows the first rows at the top).

4.2.1 Single Channel

Following is an excerpt from a configuration file which contains the keywords used for deinterlacing a single channel test image, Top-Bottom (`FRAMEMODE=0`), Left-Right (`AD1L`). Note that keywords need not appear in any particular order in the configuration file. The first pixel appears at the top left.

```
FRAMEMODE=0
TAPLINE0="AD1L,1.0,100"
TAPLINES=1
PIXELCOUNT=3072
LINECOUNT=3080
```

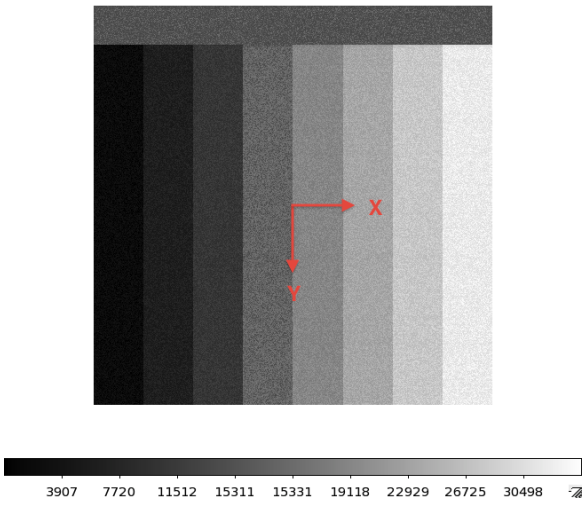

Figure 2: Test frame from example § 4.2.1. The first transmitted rows are the 300 single-valued rows and the first transmitted columns are the darkest shade. Deinterlacing is Top-to-Bottom (`FRAMEMODE=0`), Left-to-Right (`AD1L`).

4.2.2 Readout Direction

To illustrate the effect of the readout direction, the same test pattern was read using `FRAMEMODE=1/AD1L` and `FRAMEMODE=0/AD1R`, shown in Figures 3 and 4, respectively. Pixel 0,0 is at top left.

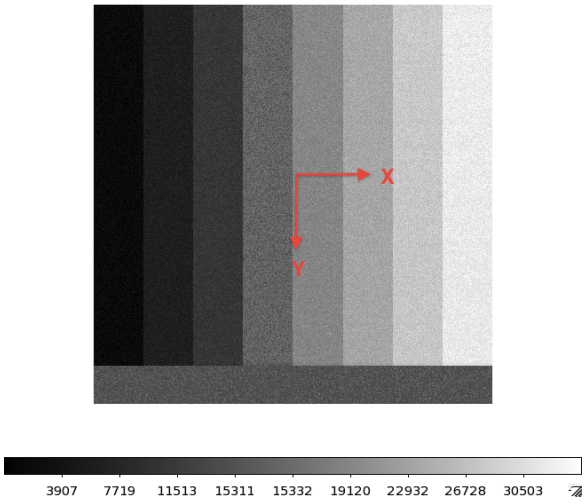

Figure 3: Deinterlacing is Bottom-to-Top (`FRAMEMODE=1`), Left-to-Right (`AD1L`).

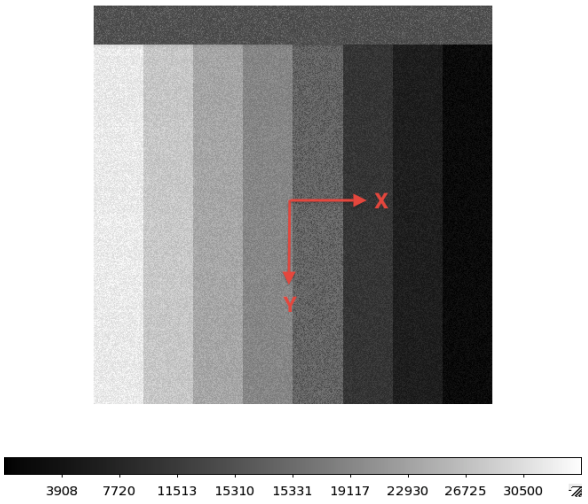

Figure 4: Deinterlacing is Top-to-Bottom (`FRAMEMODE=0`), Right-to-Left (`AD1R`).

4.2.3 Four Channel, Top Left

Following is an excerpt from a configuration file which contains the keywords used for deinterlacing a 4-channel image, 3072×3080 pixels per channel. FRAMEMODE is unchanged to show the effect of Top deinterlacing, and the direction is "L" for each tap. The gains were changed to highlight the location of each tap in the resultant image, shown in Figure 5.

```
FRAMEMODE=0
TAPLINE0="AD1L,1.0,100"
TAPLINE1="AD2L,0.7,100"
TAPLINE2="AD3L,0.4,100"
TAPLINE3="AD4L,0.1,100"
TAPLINES=4
PIXELCOUNT=3072
LINECOUNT=3080
```

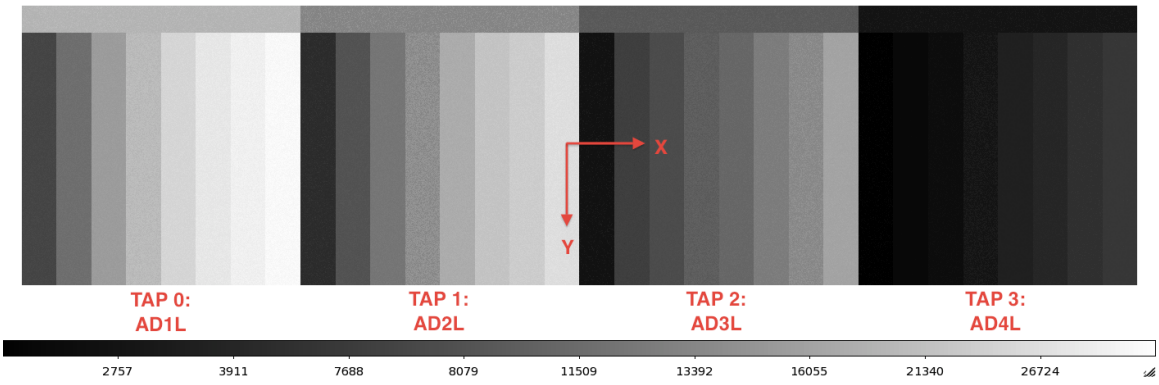

Figure 5: Resultant image from example § 4.2.3. Note the intensity change with each tap as a result of the differing gains applied, in order to correlate a TAPLINE with a given area of the final image.

4.2.4 Four Channel, Split

In the previous example of § 4.2.3, all four channels were read Top Left. This example shows the result of using the Split deinterlacing mode (`FRAMEMODE=2`), again for a 4-channel image, 3072×3080 pixels per channel. The resultant image is shown in Figure 6. Comparing the images in Fig. 5 and Fig. 6 one notices that the same shade level is associated with the same tap number in each case; that is, the brightest (gain=1) on tap0, leading to the darkest (gain=0.1) on tap3. This is expected because the same gain is assigned to the same tap numbers (channels) in each case, via the `ADnd` string; the only thing that has changed is the FRAMEMODE.

```
FRAMEMODE=2
TAPLINE0="AD1L,1.0,100"
TAPLINE1="AD2L,0.7,100"
TAPLINE2="AD3L,0.4,100"
TAPLINE3="AD4L,0.1,100"
TAPLINES=4
PIXELCOUNT=3072
LINECOUNT=3080
```

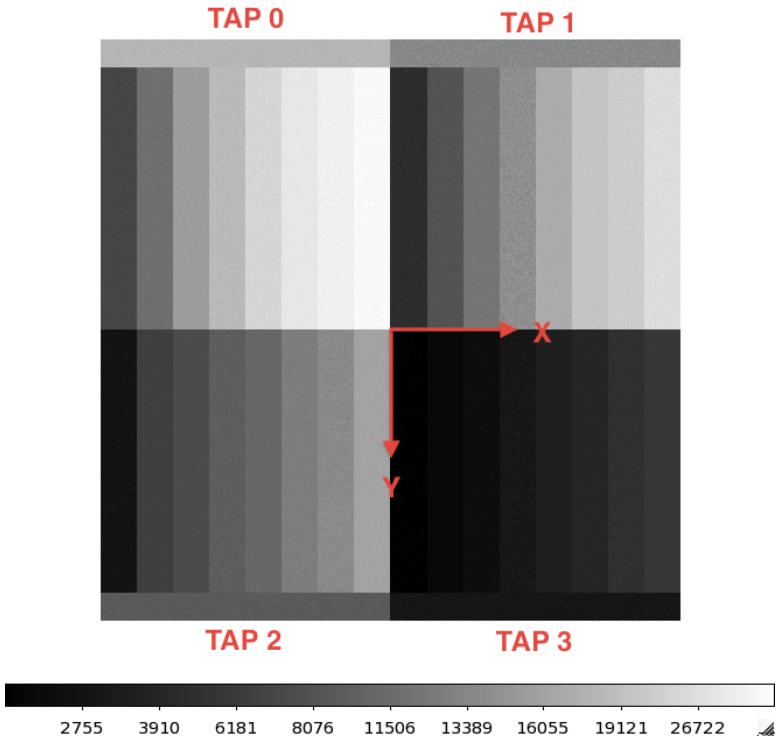

Figure 6: Resultant image from example § 4.2.4. This is the Split frame mode, where the first sampled pixels from the first half of the taps (i.e., 0 and 1) are written to the top of the frame buffer and the first sampled pixels from the second half of the taps (i.e., 2 and 3) are written to the bottom of the frame buffer.

4.3 CCD 231-C6

ZTF will use the e2v CCD231-C6 device, shown schematically in Figure 7. The CCD231 can be operated in a full-frame or split full-frame mode with readout through one, two or four amplifiers. The split full frame readout mode through four amplifiers to be used by ZTF is identified by the red arrows in Figure 7.

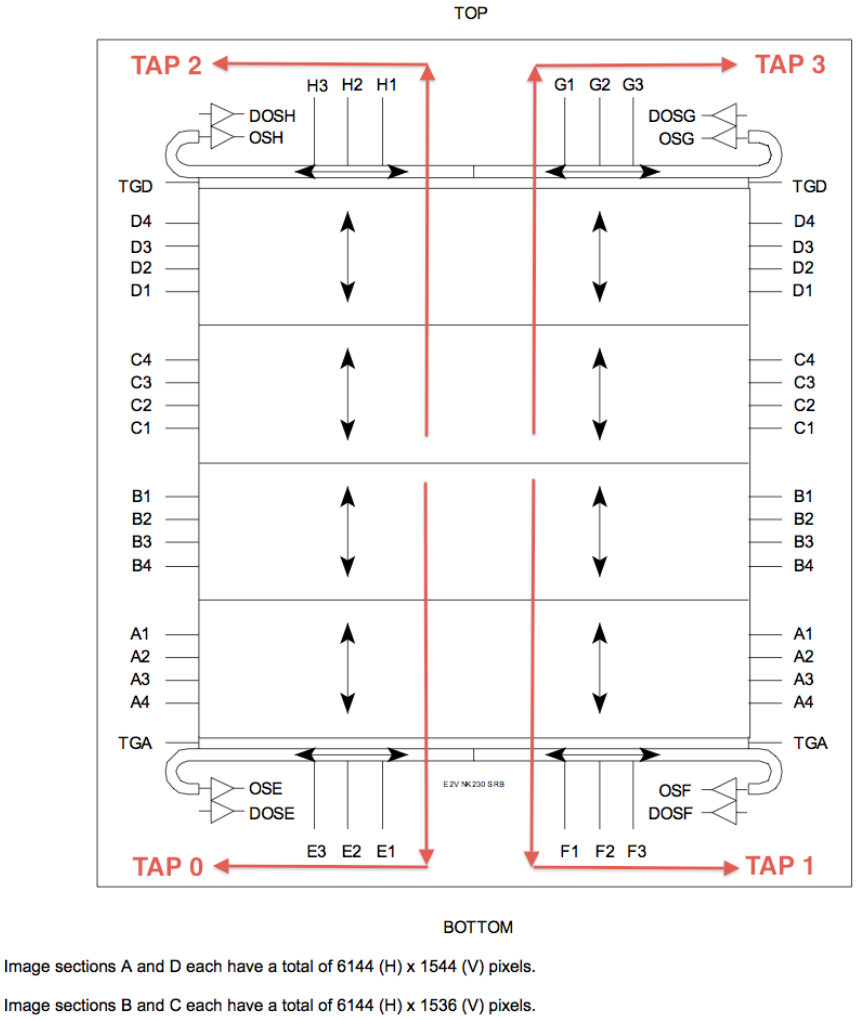

Figure 7: CCD231-C6 Schematic. Split full frame readout mode through four amplifiers is shown in red.

The Archon deinterlacing pattern which is complementary to CCD231 is `FRAMEMODE=2` (i.e., Split) and alternating every other tap between Left and Right. This is achieved through the configuration shown below and illustrated in Figure 8, where the Y axis orientation has *not* been reversed, i.e. this is the FITS image standard display orientation.

The Y axis is shown in standard orientation so that one might compare this test pattern to the CCD231 layout shown in Figure 7. The bottom left of the CCD (amplifier E) will connect to tap 0; bottom right (amplifier F) will connect to tap 1. Since tap 2 is deinterlaced to the upper right, we'll have to connect CCD amplifier H to tap 2; likewise, amplifier G will connect to tap 3.

As can be seen in Figure 7, amplifiers E and H (taps 0 and 2, respectively) will need to be read out left-to-right, and amplifiers F and G (taps 1 and 3, respectively) will be read out right-to-left. The taps have been chosen thusly, 0 and 1 for the bottom and 1 and 3 for the top. The lower left corner of the CCD will be pixel 0,0.

```
FRAMEMODE=2
TAPLINE0="AD1L,1.0,100"
TAPLINE1="AD2R,0.7,100"
TAPLINE2="AD4L,0.4,100"
TAPLINE3="AD3R,0.1,100"
TAPLINES=4
PIXELCOUNT=3072
LINECOUNT=3080
```

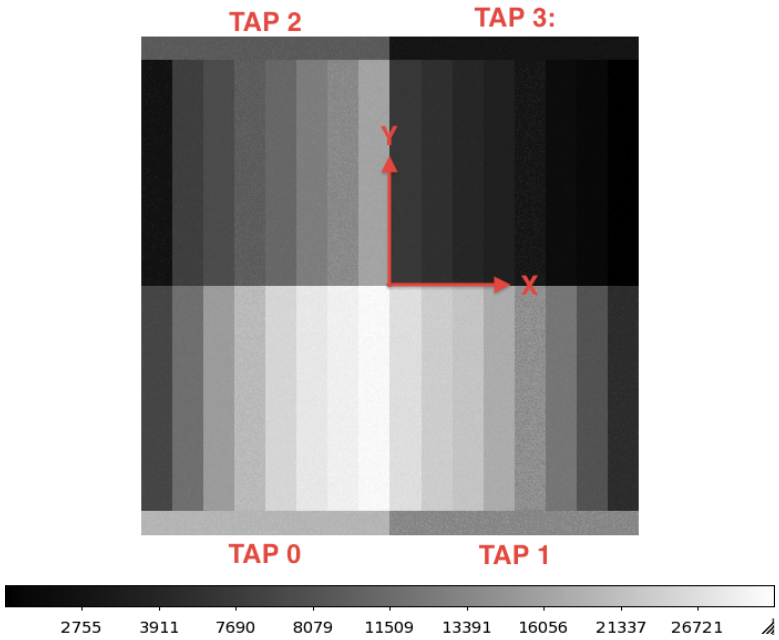

Figure 8: Test image read in split frame mode with tap readout direction alternating between Left and Right. Note the red axis labels; this is displayed in the conventional FITS image orientation with 0,0 at lower left.

4.4 Multiple ADC Modules (video boards)

Sampling multiple ADC (video) Modules is as simple as defining a tapline for each channel to be read. To delinterlace these additional channels, recall that for Split/FRAMEMODE=2, the first half of the taps are written to the top of the frame, and the second half are written to the bottom of the frame (see § 4.1 and also Figure 6). To understand how this effects the TAPLINE and AD channel configuration, consider the arrangement of CCDs depicted in the middle panel of Figure 9 (the top panel is the case already described in section § 4.2.4). Since the first half of the taps are written to the top of the frame, this means that the first four taps are going to cross over into the next physical CCD; thus, tap numbers are not contiguous for a given device. The bottom panel of Figure 9 depicts the situation for a completely populated Archon controller, four ADC modules reading four CCD231s.

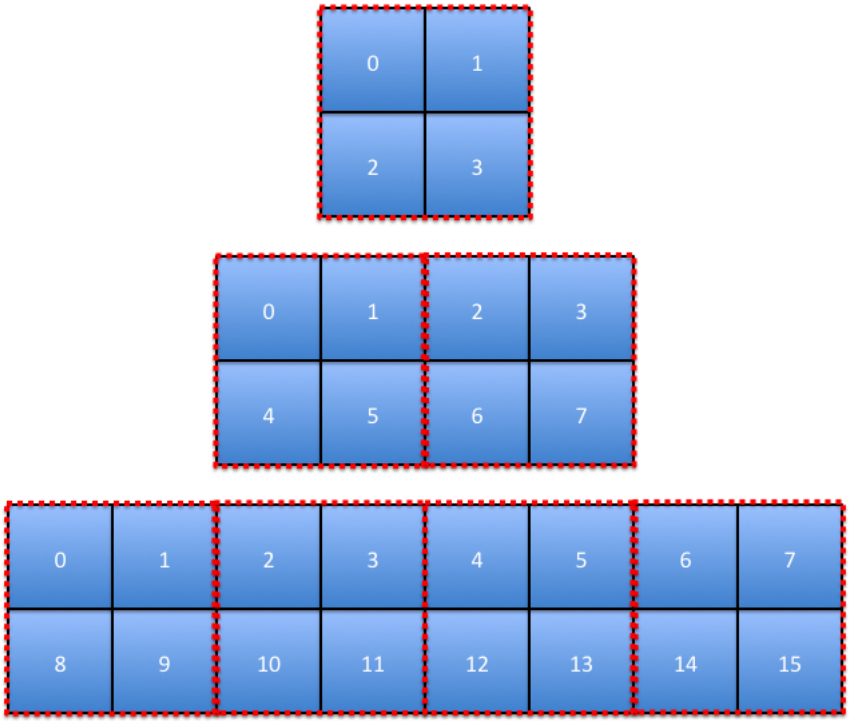

Figure 9: The top panel shows the tap order for deinterlacing four channels on one ADC Module. The middle panel shows the new tap order when a second ADC Module is added. The bottom panel is the case for four CCDs and four ADC Modules, the configuration for ZTF. Each blue box represents the region read by a particular tap, with the number indicating the tap number. The broken red lines outline the area of a single CCD.

The previous sections have shown test results from a single ADC Module (top panel Figure 9). The following sections will show test results from two and four ADC Modules, the middle and bottom panel of Figure 9, respectively. The test pattern previously described was fed into the four inputs of one ADC Module; the other ADC Module inputs were left floating. Offsets were changed for the additional modules so that the placement effects of deinterlacing configuration can be seen.

4.4.1 Two ADC Modules

The following configuration file was used for testing two ADC Modules:

```
FRAMEMODE=2
TAPLINE0="AD1L,1.0,100"
TAPLINE1="AD2R,0.7,100"
TAPLINE2="AD5L,0.5,200"
TAPLINE3="AD6R,0.5,300"
TAPLINE4="AD3L,0.4,100"
TAPLINE5="AD4R,0.1,100"
TAPLINE6="AD7L,0.5,600"
TAPLINE7="AD8R,0.5,700"
TAPLINES=8
PIXELCOUNT=3072
LINECOUNT=3080
```

The image produced by this configuration is shown in Figure 10. Compare this to the middle panel of Figure 9.

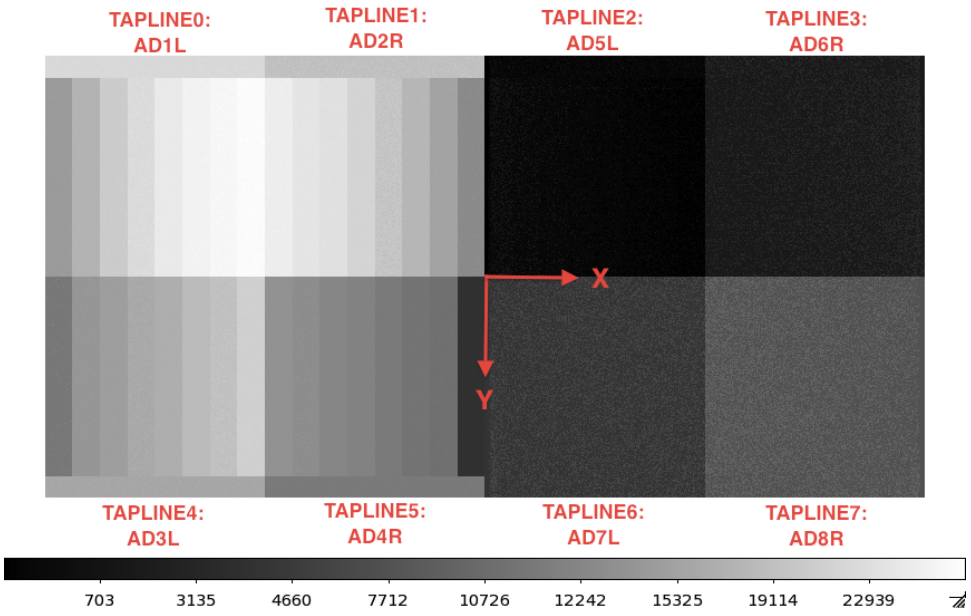

Figure 10: Actual test data from deinterlacing eight channels (taps) from two ADC Modules (i.e., video boards). The Y axis has been inverted from conventional FITS in order to compare better to the Archon nomenclature. ADC offsets are used to identify outputs with their corresponding TAPLINE configuration definition. Offset value increases from 200 (darkest) to 700 (brightest). Compare this also to Figure 9.

4.4.2 Four ADC Modules

The configuration for the final Archon controller for ZTF should look something like the following, which is also defined in Table 1. A test image is shown in Figure 11.

```
FRAMEMODE = 2
TAPLINES  = 16
PIXELCOUNT= 3072
LINECOUNT = 3080
TAPLINE0  = "AD1L,  1.0, 100"
TAPLINE1  = "AD2R,  1.0, 100"
TAPLINE2  = "AD5L,  1.0, 100"
TAPLINE3  = "AD6R,  1.0, 100"
TAPLINE4  = "AD9L,  1.0, 100"
TAPLINE5  = "AD10R, 1.0, 100"
TAPLINE6  = "AD13L, 1.0, 100"
TAPLINE7  = "AD14R, 1.0, 100"
TAPLINE8  = "AD4L,  1.0, 100"
TAPLINE9  = "AD3R,  1.0, 100"
TAPLINE10 = "AD8L,  1.0, 100"
TAPLINE11 = "AD7R,  1.0, 100"
TAPLINE12 = "AD12L, 1.0, 100"
TAPLINE13 = "AD11R, 1.0, 100"
TAPLINE14 = "AD16L, 1.0, 100"
TAPLINE15 = "AD15R, 1.0, 100"
```

CCD	Amplifier	*tap*	*ADC chan*
1	E	0	1
	F	1	2
	G	9	3
	H	8	4
2	E	2	5
	F	3	6
	G	11	7
	H	10	8
3	E	4	9
	F	5	10
	G	13	11
	H	12	12
4	E	6	13
	F	7	14
	G	15	15
	H	14	16

Table 1: Proposed wiring for a single Archon controller with four ADC Modules to read four CCDs.

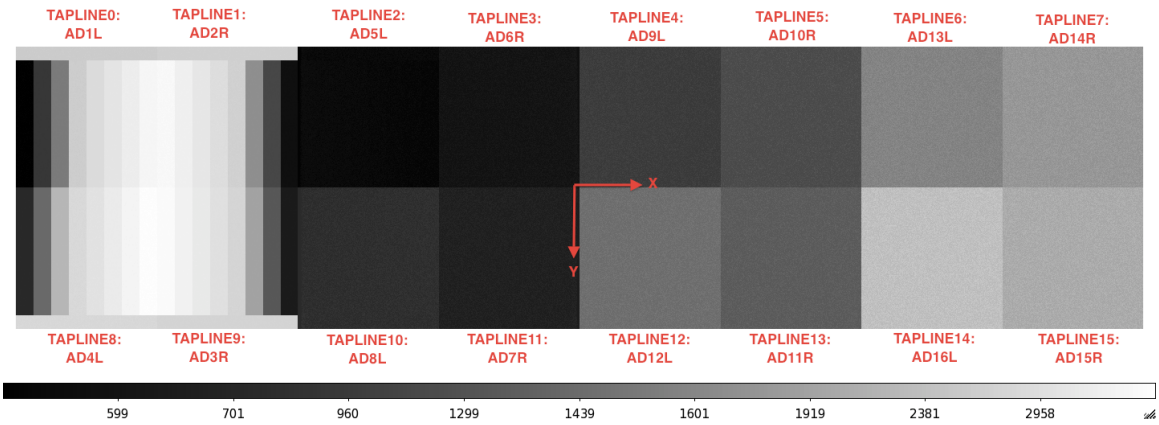

Figure 11: Actual test data from deinterlacing all 16 channels (taps) from four ADC Modules (i.e., video boards). The Y axis has been inverted from conventional FITS in order to compare better to the Archon nomenclature. ADC offsets are used to identify outputs with their corresponding TAPLINE configuration definition. Compare this also to Figure 9.

5 Post-Archon Deinterlacing

Four Archon X12 controllers will be required, each with four ADC (video board) modules, to read the 16 CCDs in the ZTF instrument. Synchronizing these four controllers is still TBD (Steve Kaye), but suffice it to say that any one controller is going to produce a single image file from the four CCDs which it controls. On the other hand, the ZTF data pipeline will require a separate FITS file for each of its 16 CCDs. In order to meet that requirement, the single image produced by each controller will need to be split into four separate FITS files. This can be simplified by ordering the frames into a reasonable pattern using the existing deinterlacing capabilities of the Archon controller. The simplest output possible is to arrange the buffers so that a single CCD appears in the correct deinterlaced pattern and store each CCD adjacent to each other. An actual test image of this pattern is shown in Figure 11.

It should be relatively simple for the host software to split the image acquired from each controller into four separate FITS files.

6 Data Compression

RICE data compression will be used when saving the data into FITS files. Routines for performing the data compression are already built into the cfitsio libraries. It is expected that ZTF will utilize the Robo-AO software, which uses CCfitsio, the C++ version of cfitsio. This should be a simple matter of calling

```
CCfits::FITS::setCompressionType ( int compType )
```

where `compType` is `RICE_1` (a defined constant). Tests have shown a savings of a factor of 10 (1.2GB to 126MB) for a single 16-channel test image using RICE compression.