IUCAA Focal Plane Array Controller (IFPAC)
Readout waveform generation scheme

Mahesh Burse, Sakya Sinha, Monali Sinare
Sep 11, 2013

Purpose of this document

The purpose of this document is to describe the concept and procedure for generating readout waveforms for a given detector and mode of operation.

IFPAC System Overview

IFPAC system is developed at IUCAA for controlling wide variety of astronomical detectors like CCD, EMCCD, CMOS, and IR etc. IFPAC is a multi-controller, multi-interface scalable system, capable of controlling all above mentioned detectors. Particular configuration of the controller can be chosen depending on the type and number of detectors to be controlled. For two four output detectors, basic configuration consisting of only one analog, one digital and one backplane card would be required. Xilinx make Virtex-5 series FPGA is used as an embedded controller for generating detector and observation specific readout clocks, reading multiple 16 bit serial ADCs and programming DACs for bias voltages. Each video channel can be read out at 1 MSPS, and is processed with analogue CDS operation and then converted to digital numbers using 16 bit serial ADCs. USB 2.0 as well as Ethernet port has been provided as PC interface for host communication and one can use either of the one at a given time.

Functionality and capability of this basic configuration can be doubled by combining two such controllers, in which case one of the digital cards acts as a master and communicates to other through fast serial Rocket-IO interface over a fiber link. For Mosaic, this system is expanded by adding a dedicated communication card to communicate and sync with multiple controllers through Rocket-IO interfaces over a fiber link. Host software runs on a Linux PC and user can configure the controller on the fly for the current detector system and demand arbitrary ROI readouts along with over scan and dark pixels.

Configuring IFPAC as per your detector / application

Based on the availability of different kind of detectors in the market as of today, we have classified them in two categories:

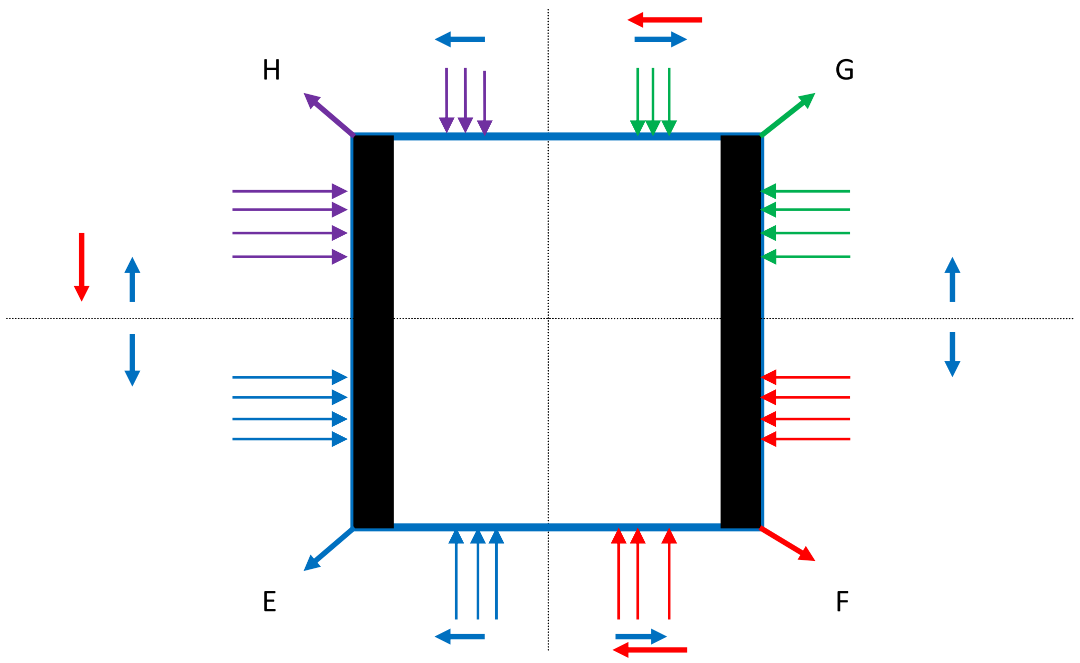
1. 1 ,2 and 4 Output Detectors
2. More than 4 output (e.g. STA or HxRG through SIDECAR interface) Detectors

To generate ROI readout clocks, IFPAC needs to know some basic information related to the detector intended to use. All this information would be part of the detector data sheet and has to be captured and passed on to IFPAC in a specified format. This is a onetime process for a given detector. User would be able give information through a parameter file values. IFPAC host software would gather this information from parameter file and send it to FPGA by way of generating and sending respective commands.

Detector specific (One time settable) parameters:

1. ActivePixelsPerLine – Active Pixels per Line for a given detector
2. LinesPerFrame – Number of Lines (or Rows) of active pixels for a given detector
3. DarkPixels – Dark Pixels per output for a given detector
4. Single or Dual Clocks -> for a class 1 detectors (i.e. 1, 2 or 4 output), w.r.t. figure below IFPAC assumes that 16 bit state table downloaded into memory is for "E" output. State tables are described in next section.
   If applying exactly same Line and serial clocks to clocks related to all other outputs results in a 4 output readout, then this is Single Clock type detector and if this results in a single (E) output readout, then this is a Dual Clock type detector.

Please note that both types of detectors are available in the market and therefore user need to select the appropriate option for this in the parameter file.

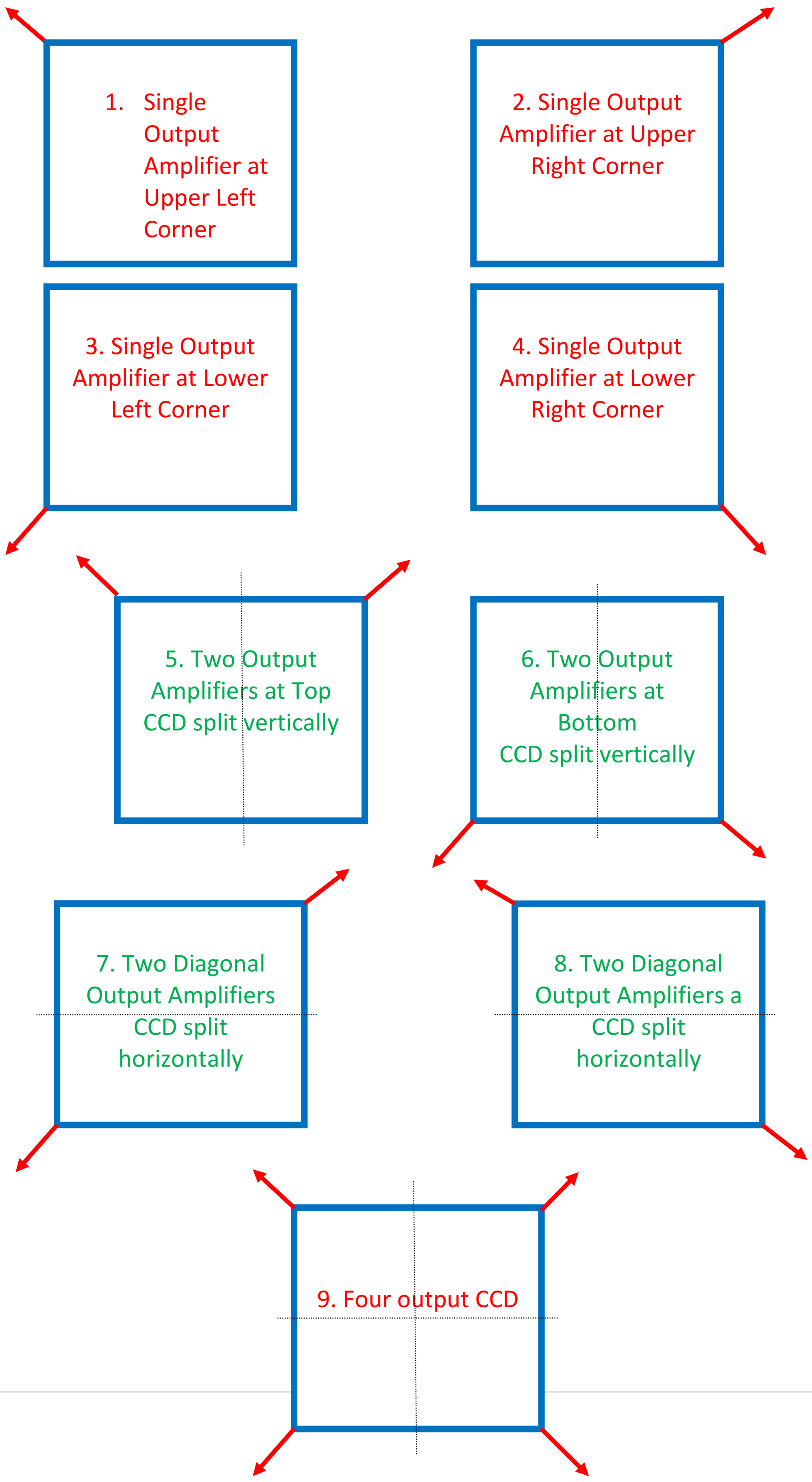


Blue arrow shows the direction of charge transfer for Single clock operation and red arrow shows the direction of charge transfer for Dual clock operation.

There is a separate field in the parameter file for LINE or PIXEL clocks for selecting Single or Dual clock type operation. Each one can be independently selected for Single / Dual operation.

5. ClockSwap -> 16 bit state table has only 4 line clocks and 3 serial clocks, but to change the direction of readout, some clock bits within LINE and SERIAL clocks need to be swapped. This is again detector specific and not same across all detector types and vendors. Therefore user has to look at the data sheet and note down which clock bits need to be swapped to change or reverse the direction of readout. Again this can be specified in a parameter file with respective parameter values.
   There is a separate field in the parameter file for swapping LINE or PIXEL clocks.

Exposure specific parameters: Selecting a detector output(s) for readout for
1, 2, and 4 output detectors
Choose the outputs from which you wish to read your detector from following options:

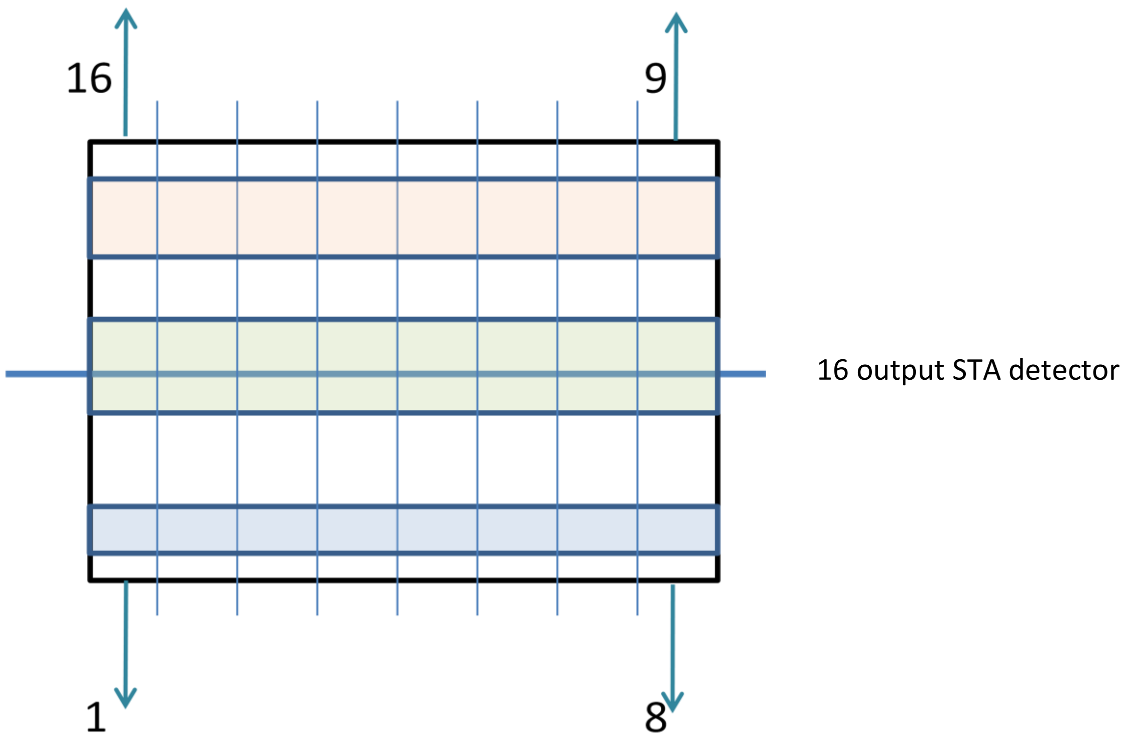

1. Single Output Amplifier at Upper Left Corner
2. Single Output Amplifier at Upper Right Corner
3. Single Output Amplifier at Lower Left Corner
4. Single Output Amplifier at Lower Right Corner
5. Two Output Amplifiers at Top CCD split vertically
6. Two Output Amplifiers at Bottom CCD split vertically
7. Two Diagonal Output Amplifiers CCD split horizontally
8. Two Diagonal Output Amplifiers a CCD split horizontally
9. Four output CCD

Please note that even if you have a 4 output detector but wish to read the data from only 1 or 2 outputs, it is possible, just select the desired output mode from the above.

In the full grown version of IFPAC with GUI, user would be able to select the desired detector output configuration by clicking on the picture. For the time being, user can select the configuration by selecting parameter value (from 1 to 9) in the parameter file.

Category 2: More than 4 output detectors

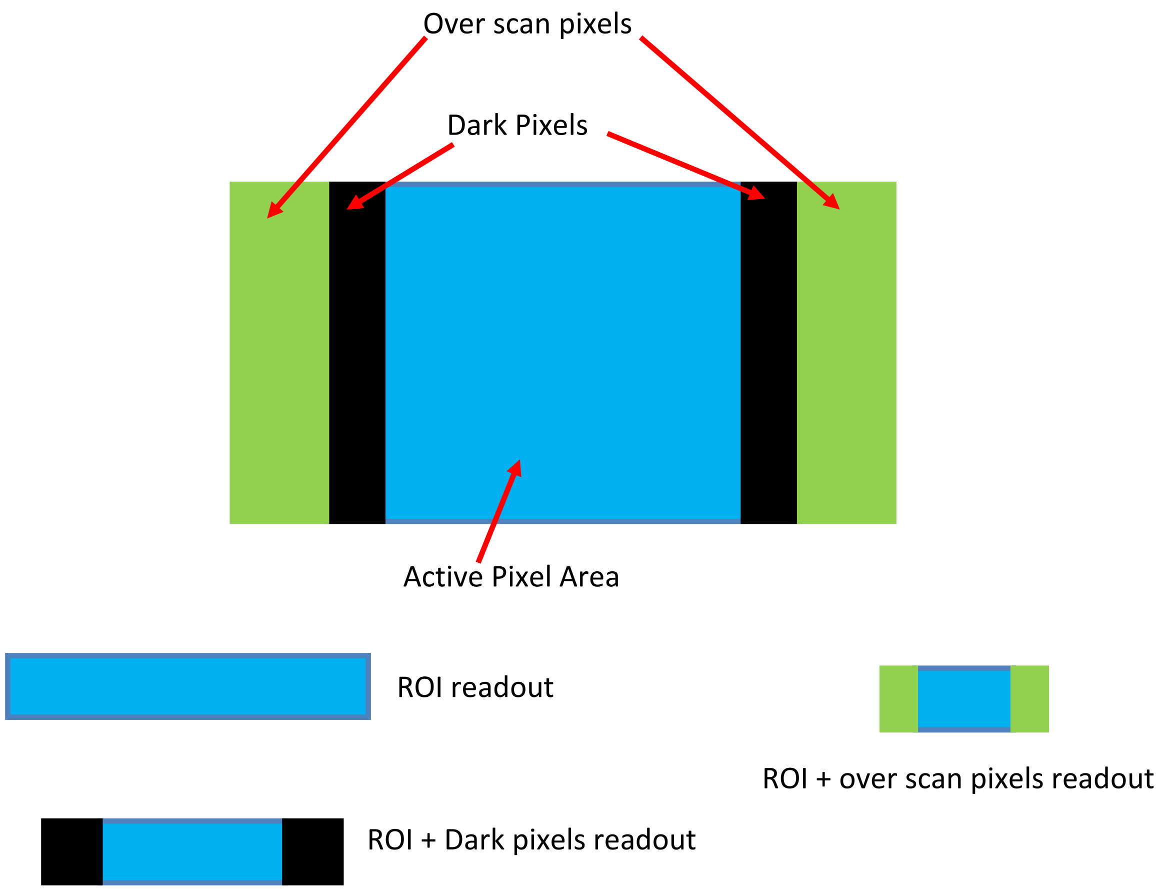
These are quite different compare to up to four output detectors. As an example, we have implemented readout scheme for STA type detector. STA as shown below is a 16 output detector and IFPAC can be used for these types of detectors with ROI readouts.



For Teledyne HAWAII detectors, IFPAC has a dedicated interface to communicate with SIDECAR ASIC. IFPAC would be completely compatible with Teledyne supported HxRG readout modes. IFPAC uses a serial interface to configure and program SIDECAR ASIC and 16 bit parallel interface for science data readout. Actual Readout and Reset clock generation is done in the ASIC.

Dark and Over-scan Pixels ->

IFPAC allows the user to choose number of dark and over-scan pixels (per output) to read for a given exposure. This implies that for ROI readout, user shall have an option to have or not to have dark and over-scan pixels in the final image.



IFPAC MODES

A single clock card (or FPGA) in an IFPAC can be set in single detector mode to generate 32 readout clocks. In this mode, use shall have lot of flexibility in terms of choosing the output(s) of a given detector for readout. In the second, multi detector mode IFPAC can generate readout clocks for two detectors simultaneously (2 X 16) but with limited flexibility in terms of choosing the output(s) of a given detector. User need to define only 16 bit clocks, which are automatically mapped to 32 bit format for single detector mode.
IFPAC has a built in intelligence to generate clocks necessary to read ROI image based on the detector size and an output of the detector from which image data is to be read out. User has to provide only very basic information (like small line and pixel state tables) and that too only once.

Process of Waveform Table Generation

User needs to make four waveform tables per detector (as per detector specs), namely

Line Transfer Table
Pixel Transfer Table
Partial Pixel Table – to fast skip some pixels
Line Dump Table – to quickly clear the CCD or to fast skip lines

Waveform tables are specific to the detector in operation and have to be generated and optimized by the user for best performance. FPGA has 1 K Word memory per waveform table, which means one can have up to 512 entries (state-time pair) / table.

- **Formation of Waveform table**

  Every table contains a waveform state in the form of 16 bit word, followed by another 16 bit word which gives time for which given state has to be held. Hence width of the table would be 16 bit but depth will depend upon number of states in the waveform. When the three phase line clock detector is in use, fourth line clock stays at one fixed state (could be ignored).

  Below is the format of 16 bit waveform state

S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
EMR	Hold	CNV	RST	SN	TGA	DG	SW	RØ	RØ3	RØ2	RØ1	I4	I3	I2	I1

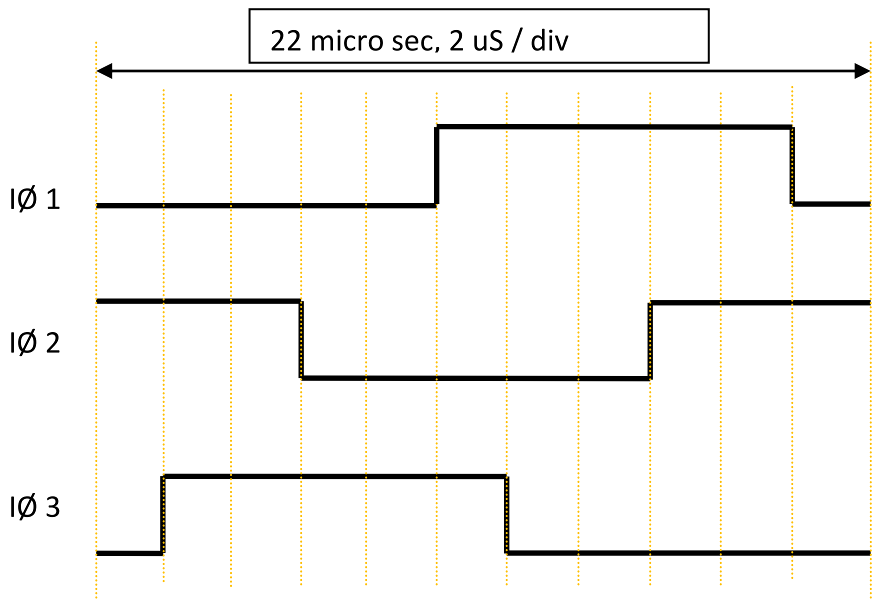
I1 to I4 (S3 to S0)	<= Line clocks
R01 to R03 (S6 to S4)	<= Serial clocks
SW (S8)	<= Summing Well
DG (S9)	<= Dump Gate
TGA (S10)	<= Transfer Gate
SN (S11)	<= Sample
RST (S12)	<= Reset
CNV (S13)	<= ADC conversion singnal
Hold (S14)	<=Hold / integration time
EMR (S15)	<=High Voltage Clock for EMCCD

Time for which particular state remains stable is to be calculated using 10 ns as time unit. e.g. Enter hex x”00C8” (200 decimal) for time 2 micro sec.
Using time field we can format output waveform with the accuracy of 10 ns. Minimum time possible for state to state transition is 30 nS. Host software shall accept numbers from 3 to 65535 (0xFFFF) for time field.

➢ Example waveform configuration table with respective waveform

• Line transfer Table

As an example, to generate a line transfer waveform table for E2V CCD 4240, look at the line clocks in the data sheet (or draw it), now keeping rest of clocks at (default) state necessary during line transfer, write down the 16 bit state values and their state time as line clocks make transitions.



Line transfer Waveform: E2V CCD 4240

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
EMR	Hold	CNV	RST	SN	TGA	DG	SW	RØ	RØ3	RØ2	RØ1	I4	I3	I2	I1	Time	Count (Hex)
0	1	0	0	1	0	0	0	1	0	1	1	0	0	1	0	2µS	C8
												0	1	1	0	4µS	190
												0	1	0	0	4µS	190
												0	1	0	1	2µS	C8
4 (h)				8 (h)				B (h)				0	0	0	1	4µS	190
												0	0	1	1	4µS	190
												0	0	1	0	2µS	C8

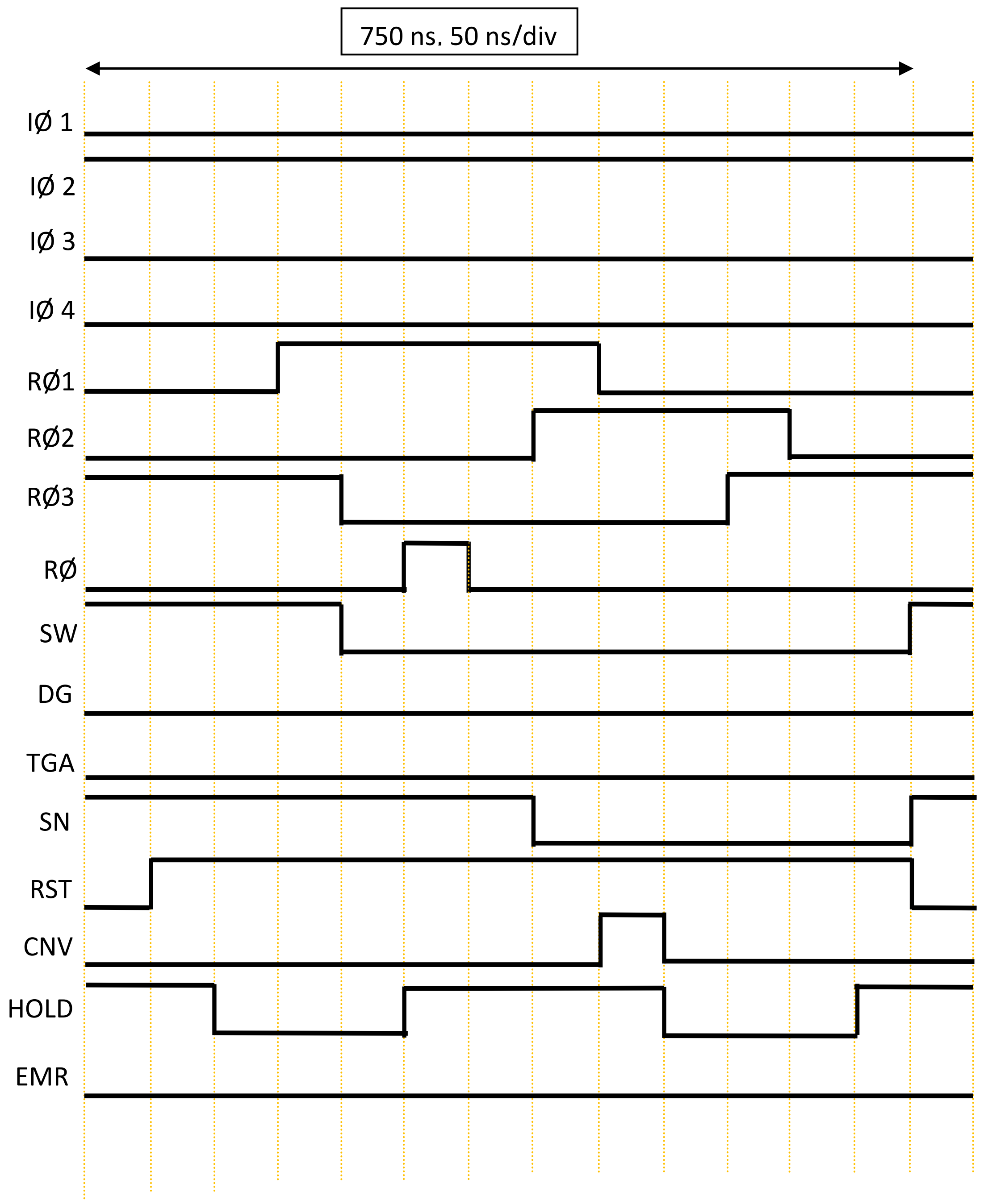
Table 1: Line transfer table: E2V CCD 4240

/// Line transfer table for above example

```
 "48B2"                    // state 1
 "00C8"                    // time for which state 1 to remain stable
"48B6"                     // state 2
"0190"                     // time for which state 2 to remain stable
"48B4"                     // state 3
"0190"                     // time for which state 3 to remain stable
"48B5"                     // state 4
"00C8"                     // time for which state 4 to remain stable
"48B1"                     // state 5
"0190"                     // time for which state 5 to remain stable
"48B3"                     // state 6
"0190"                     // time for which state 6 to remain stable
"48B2"                     // state 7
"00C8"                     // time for which state 7 to remain stable
```

- **Pixel transfer Table**

As an example, to generate a Pixel transfer waveform table for E2V CCD 4240, look at the pixel clocks in the data sheet (or draw it) and write down the 16 bit state values and their state time as line clocks make transitions.



Pixel transfer Waveform: E2V CCD 4240

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
EMR	Hold	CNV	RST	SN	TGA	DG	SW	RØ	RØ3	RØ2	RØ1	I4	I3	I2	I1	Time	Count (Hex)
0	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	50ns	05
0	1	0	1	1	0	0	1	0	1	0	0					50ns	05
0	0	0	1	1	0	0	1	0	1	0	0					50ns	05
0	0	0	1	1	0	0	1	0	1	0	1					50ns	05
0	0	0	1	1	0	0	0	0	0	0	1					50ns	05
0	1	0	1	1	0	0	0	1	0	0	1					50ns	05
0	1	0	1	1	0	0	0	0	0	0	1					50ns	05
0	1	0	1	0	0	0	0	0	0	1	1	2 (h)				50ns	05
0	1	1	1	0	0	0	0	0	0	1	0					50ns	05
0	0	0	1	0	0	0	0	0	0	1	0					50ns	05
0	0	0	1	0	0	0	0	0	1	1	0					50ns	05
0	0	0	1	0	0	0	0	0	1	0	0					50ns	05
0	1	0	1	0	0	0	0	0	1	0	0					150ns	F
0	1	0	0	1	0	0	1	0	1	0	0						

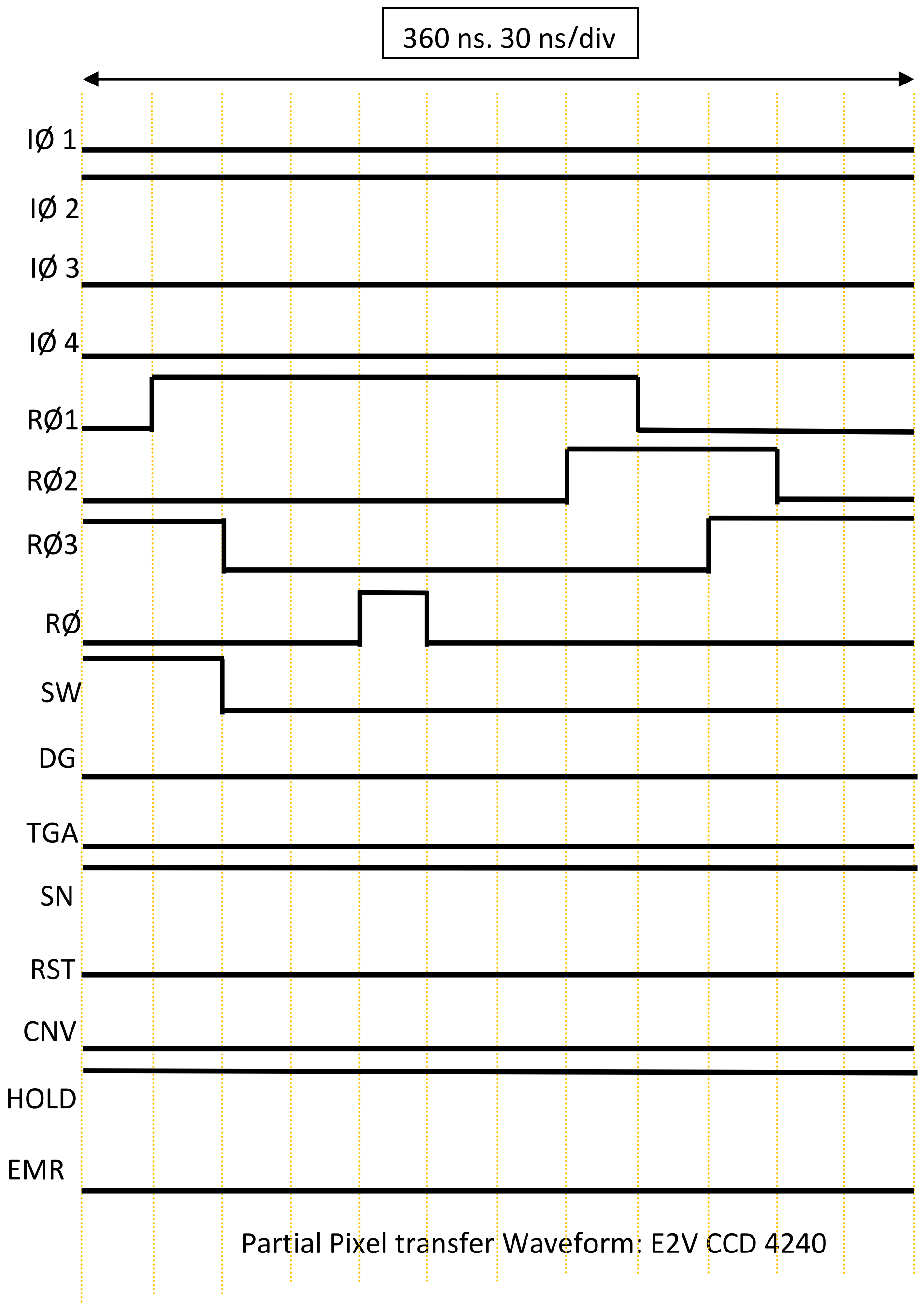
Table 2: Pixel transfer table: E2V CCD 4240

```
/// Pixel transfer table for above example

     "4942"                  // state 1
     "0005"                  // time for which state 1 to remain stable
    "5942"                   // state 2
    "0005"                   // time for which state 2 to remain stable
    "1942"                   // state 3
    "0005"                   // time for which state 3 to remain stable
    "1952"                   // state 4
    "0005"                   // time for which state 4 to remain stable
    "1812"                   // state 5
    "0005"                   // time for which state 5 to remain stable
    "5892"                   // state 6
    "0005"                   // time for which state 6 to remain stable
    "5812"                   // state 7
    "0005"                   // time for which state 7 to remain stable
    "5032"                   // state 8
     "0005"                  // time for which state 8 to remain stable
    "5022"                   // state 9
    "0005"                   // time for which state 9 to remain stable
    "1022"                   // state 10
    "0005"                   // time for which state 10 to remain stable
    "1062"                   // state 11
    "0005"                   // time for which state 11 to remain stable
    "1042"                   // state 12
    "0005"                   // time for which state 12 to remain stable
    "5042"                   // state 13
    "0005"                   // time for which state 13 to remain stable
    "4942"                   // state 14
    "000F"                   // time for which state 14 to remain stable
```

- **Partial Pixel transfer Table**

As an example, to generate a Partial Pixel transfer waveform table for E2V CCD 4240, look at the partial pixel clock waveform in the data sheet (or draw it) and write down the 16 bit state values and their state time as line clocks make transitions.



Partial Pixel transfer Waveform: E2V CCD 4240

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
EMR	Hold	CNV	RST	SN	TGA	DG	SW	RØ	RØ3	RØ2	RØ1	I4	I3	I2	I1	Time	Count (Hex)
0	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	30ns	03
				1	0	0	1	0	1	0	1					30ns	03
				1	0	0	0	0	0	0	1					60ns	06
				1	0	0	0	1	0	0	1					30ns	03
4 (h)				1	0	0	0	0	0	0	1	2 (h)				60ns	06
				1	0	0	0	0	0	1	1					30ns	03
				1	0	0	0	0	0	1	0					30ns	03
				1	0	0	0	0	1	1	0					30ns	03
				1	0	0	0	0	1	0	0					30ns	03
				1	0	0	0	0	1	0	0					30ns	03

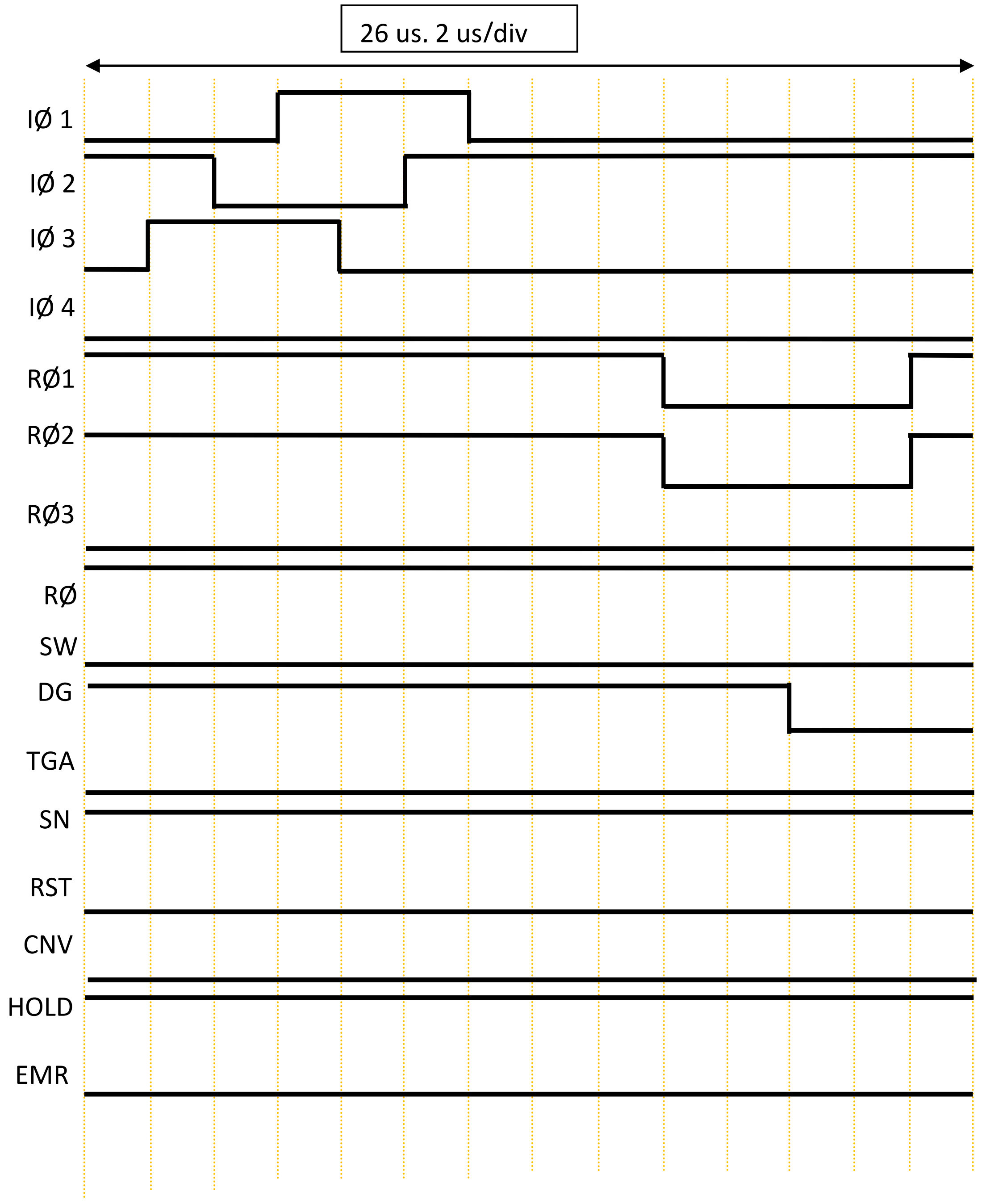
Table 3: Partial Pixel transfer table: E2V CCD 4240

```
/// Partial Pixel transfer table for above example

        "4942"                    // state 1
         "0003"                   // time for which state 1 to remain stable
        "4952"                    // state 2
        "0003"                    // time for which state 2 to remain stable
        "4812"                    // state 3
        "0006"                    // time for which state 3 to remain stable
        "4892"                    // state 4
        "0003"                    // time for which state 4 to remain stable
        "4812"                    // state 5
        "0006"                    // time for which state 5 to remain stable
        "4832"                    // state 6
        "0003"                    // time for which state 6 to remain stable
        "4822"                    // state 7
        "0003"                    // time for which state 7 to remain stable
        "4862"                    // state 8
        "0003"                    // time for which state 8 to remain stable
        "4842"                    // state 9
        "0003"                    // time for which state 9 to remain stable
        "4842"                    // state 10
        "0003"                    // time for which state 10 to remain stable
```

- **Dump Line Table**

As an example, to generate a Dump Line waveform table for E2V CCD 4240, look at the Dump line clock waveform in the data sheet (or draw it) and write down the 16 bit state values and their state time as line clocks make transitions.



Partial Pixel transfer Waveform: E2V CCD 4240

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
EMR	Hold	CNV	RST	SN	TGA	DG	SW	RØ	RØ3	RØ2	RØ1	I4	I3	I2	I1	Time	Count (Hex)
0	1	0	0	1	0	1	0	1	0	1	1	0	0	1	0	2µS	C8
4 (h)				A (h)				B (h)				0	1	1	0	2µS	C8
												0	1	0	0	2µS	C8
												0	1	0	1	2µS	C8
												0	0	0	1	2µS	C8
												0	0	1	1	2µS	C8
												0	0	1	0	4µS	190
				1	0	1	0	1	0	0	0	0	0	1	0	4µS	190
				1	0	0	0	1	0	0	0	0	0	1	0	4µS	190
				1	0	0	0	1	0	1	1	0	0	1	0	2µS	C8

Table 4: Dump Line table : E2V CCD 4240

/// Dump Line table for above example

```
"4AB2"                              // state 1
 "00C8"                             // time for which state 1 to remain stable
"4AB6"                              // state 2
"00C8"                              // time for which state 2 to remain stable
"4AB4"                              // state 3
"00C8"                              // time for which state 3 to remain stable
"4AB5"                              // state 4
"00C8"                              // time for which state 4 to remain stable
"4AB1"                              // state 5
"00C8"                              // time for which state 5 to remain stable
"4AB3"                              // state 6
"00C8"                              // time for which state 6 to remain stable
"4AB2"                              // state 7
"0190"                              // time for which state 7 to remain stable
"4A82"                              // state 8
"0190"                              // time for which state 8 to remain stable
"4882"                              // state 9
"0190"                              // time for which state 9 to remain stable
"48B2"                              // state 10
"00C8"                              // time for which state 10 to remain stable
```

Output Clock mapping With respect to 16 bit Waveform state register

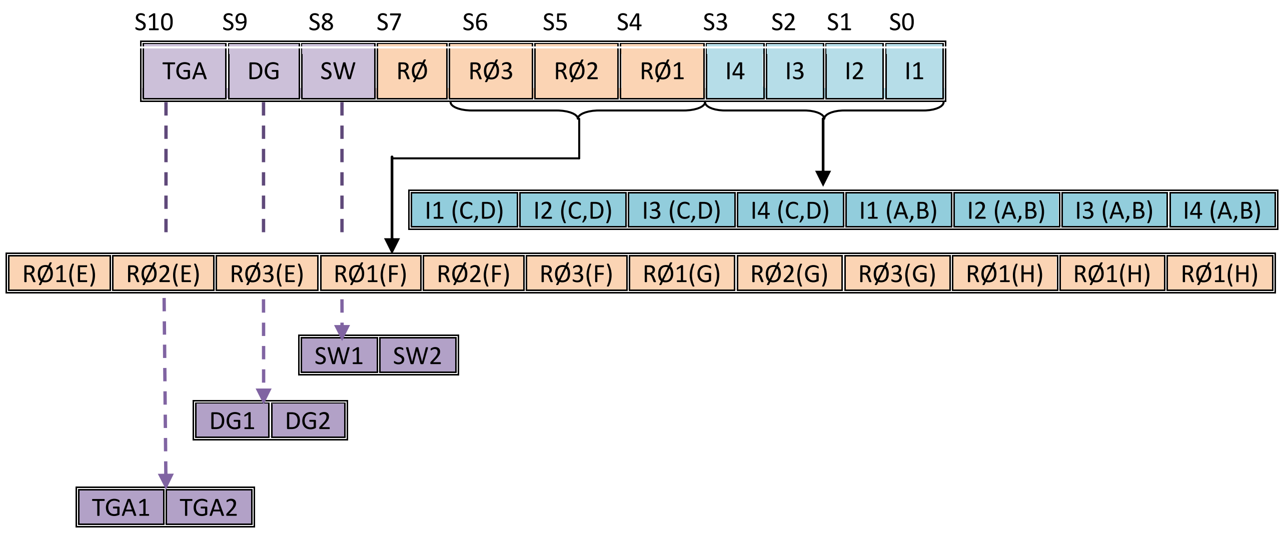
➢ **16 bit state Register**

S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
EMR	Hold	CNV	RST	SN	TGA	DG	SW	RØ	RØ3	RØ2	RØ1	I4	I3	I2	I1

I1 to I4 (S3 to S0)	<= Line clocks
R01 to R03 (S6 to S4)	<= Serial clocks
SW (S8)	<= Summing Well
DG (S9)	<= Dump Gate
TGA (S10)	<= Transfer Gate
SN (S11)	<= Sample
RST (S12)	<= Reset
CNV (S13)	<= ADC conversion singnal
Hold (S14)	<=Hold / integration time
EMR (S15)	<=High Voltage Clock for EMCCD

Signals like HOLD, CNV, RST and SN are just mapped to the corresponding pins of analog processing integrated chips on analog board through a backplane. Signals related to generation of Clock wave form like line clock, pixel clock, summing well, DG and TGA get distributed as per selected mode i.e. Single detector mode or Multi detector mode. Distribution of clock related signals is explained below.

- **Mapping for single detector mode**



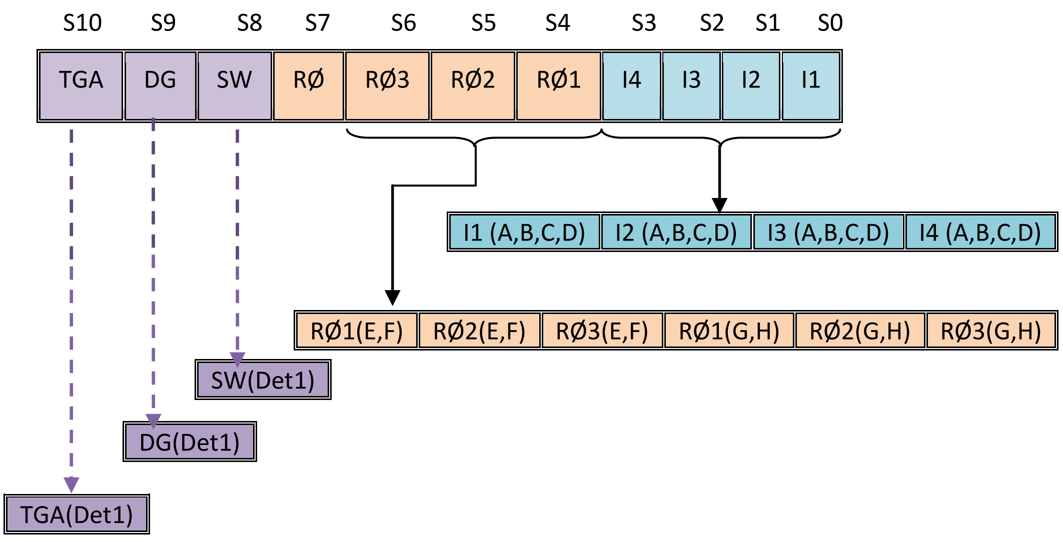
A, B, C, D: Line Clock Outputs – Two sets per detector to decide direction (Up or down) of charge transfer
E, F, G, H: Pixel Clock Outputs – One set per output assuming 4 output CCD
This is the default mapping, based on the desired output modes user can decide which outputs to gang together.
Having multiple copies of TGA, DG, SW increases driving capacity.

Clock Outputs on backplane connector for Single detector Mode

Clock No.	Output (J47)	Description
Clock1	J47.2	I1 (C, D)
Clock2	J47.17	I2 (C, D)
Clock3	J47.4	I3 (C, D)
Clcok4	J47.20	I4 (C, D)
Clock5	J47.18	I1 (A, B)
Clock6	J47.3	I2 (A, B)
Clock7	J47.19	I3 (A, B)
Clock8	J47.6	I4 (A, B)
Clock9	J47.21	RØ1 (E)
Clock10	J47.7	RØ2 (E)
Clock11	J47.9	RØ3 (E)
Clock12	J47.16	RØ1 (F)
Clock13	J47.1	
Clock14	J47.22	RØ2 (F)
Clock15	J47.5	RØ3 (F)
Clock16	J47.8	RØ1 (G)
Clock17	J47.26	RØ2 (G)
Clock18	J47.24	RØ3 (G)

Clock19	J47.11	RØ1 (H)
Clock20	J47.25	RØ2 (H)
Clock21	J47.10	RØ3 (H)
Clock22	J47.23	RØ
Clock23	J47.12	RØ
Clock24	J47.14	SW
Clock25	J47.13	SW
Clock26	J47.30	DG
Clock27	J47.44	DG
Clock28	J47.27	TGA
Clock29	J47.28	TGA
Clock30	J47.29	
Clock31	J47.15	
Clock32		EMR

- **Mapping for Multi Detector mode**



A, B, C, D: Line Clock Inputs
E, F, G, H: Pixel Clock Inputs

In this mode, there is going to be separate detector specific 16 bit waveform state register for each detector and since numbers of available clocks per detector are restricted, many clocks may have to be ganged together. This mode allows user to run two same or different detectors simultaneously sharing the same hardware.

Clock Output on backplane connector for Multi detector Mode

Clock No.	**Output (J47)**	**Description**
Clock1	J47.2	I1 (Det2)
Clock2	J47.17	I2 (Det2)
Clock3	J47.4	I3 (Det2)
Clcok4	J47.20	I4 (Det2)
Clock5	J47.18	I1 (Det1)
Clock6	J47.3	I2 (Det1)
Clock7	J47.19	I3 (Det1)
Clock8	J47.6	I4 (Det1)
Clock9	J47.21	RØ1 (Det1, E, F)
Clock10	J47.7	RØ2 (Det1, E, F)
Clock11	J47.9	RØ3 (Det1, E, F)
Clock12	J47.16	RØ1 (Det1, G, H)
Clock13	J47.1	
Clock14	J47.22	RØ2 (Det1, G, H)
Clock15	J47.5	RØ3 (Det1, G, H)
Clock16	J47.8	RØ1 (Det2, E,F)
Clock17	J47.26	RØ2 (Det2, E,F)
Clock18	J47.24	RØ3 (Det2, E,F)
Clock19	J47.11	RØ1 (Det2, G,H)
Clock20	J47.25	RØ2 (Det2, G,H)
Clock21	J47.10	RØ3 (Det2, G,H)
Clock22	J47.23	RØ (Det1)
Clock23	J47.12	RØ (Det1)
Clock24	J47.14	SW2 (Det2)
Clock25	J47.13	SW1 (Det1)
Clock26	J47.30	DG1 (Det1)
Clock27	J47.44	DG2 (Det2)
Clock28	J47.27	TGA1 (Det1)
Clock29	J47.28	TGA2 (Det2)
Clock30	J47.29	RØ (Det2)
Clock31	J47.15	RØ (Det2)
Clock32		EMR

Exposure Sequence

Exposure sequence is same irrespective of the detector and output mode, by setting the appropriate values for the sequence related parameters, user can demand any desired sequence based on application.

Parameters related to sequence

NResets ->
// Clear the entire CCD using line dump table
// For CMOS or IR, generate appropriate RESET clocks

NReads ->
// Read detector as per the co-ordinates set for this exposure

NGroups ->
// Repeat NReads frames this (NGroup) time,
// more relevant for IR detector, can be SET to 1 for CCD

NRamps ->
// Repeat the entire NReset followed by NReads * NGroup sequence

URG ->
// Up the Ramp Sampling scheme, more relevant for IR detector

FS ->
// N Fowler Sampling scheme, more relevant for IR detector

ExpTime ->
// can be specified in units of ms, range 1 ms to 9.5 hours
Example
1) Single Exposure for CCD,
NResets = 1, NReads = 1, NGroups = 1, NRamps = 1

This will clear the CCD once -> open the shutter for ExpTime value -> close the shutter -> Read the frame once
Host would receive one frame worth data as per ROI parameters.
Clocks could be monitored on oscilloscope / LA at J47 on backplane or on dummy load board as per details given later in this document.

2) Multiple Exposures for CCD
NResets = 1, NReads = 1, NGroups = 1, NRamps = 100
This will repeat the sequence specified for earlier example 100 times.
Host would receive 100 frames of data as per ROI parameters.
Clocks could be monitored on oscilloscope / LA at J47 on backplane or on dummy load board as per details given later in this document.

ShutterEnable ->
IFPAC has one shutter control line per detector and if user selects this option then respective shutter control line will stay high for the time equal to ExpTime.

Some other parameters

SendADCData ->

Current version of utility host software does not have the capability to read arbitrary areas of the detector but one can generate and see the arbitrary clocks for ROI readout on a oscilloscope. Therefore this parameter would be used by software to switch between

a) data acquisition plus clock generation for full frame images
&
b) Arbitrary clock generation without data acquisition for all modes.

Binning ->

Binning is yet to be implemented, but basically we have two options

a) Use the necessary information from LINE and PIXEL transfer tables and repeat them X or Y bin times respectively.
b) Upload the new waveform tables (with binning embedded into it)

Sample Parameter File

```
// Parameter values to be specified only ONCE to specify the Detector
//
DetectorType            = CCD          // CCD, STA
NOutPutsCCD             = 9            // 1: 1OutputUL/2: 1OutputUR/3: 1OutputLR/
                                       // 4: 1OutputLL/5: 2OutputULLR/
                                       // 6: 2OutputURLL/7: 2OutputL/8: 2OutputU/
                                       // 9: 4OutPut
NOutputsSTA             = 1            // 1: 4Outputs/2: 8Outputs/3: 16Outputs
LineClkSingleDual       = 1            // 1: Default readout direction/
PixelClkSingleDual      = 1            // 2: Reverse readout direction
DarkPixels              = 5            // Dark Pixels per output, 8 bit value,
                                       //  range 0 to 255
ActivePixelsPerLine     = 4096         // Detector X Size,
                                       // Number of Active Pixels per Line (Row)
LinesPerFrame           = 4096         // Detector Y Size, Number of Lines
LineClkSwap             = OneWithTwo     // OneWithTwo, OneWithThree,
                                         // OneWithFour, TwoWithThree
                                        //TwoWithFour/ ThreeWithFour
PixelClkSwap            = OneWithTwo   // OneWithTwo/OneWithThree/
                                       // TwoWithThree
// Parameters which may have to be specified per Exposure for above detector
//configuration
ShutterEnable           = YES          // YES/NO
OverScanPixels          = 5            // Overscan Pixels per output,
                                               //8 bit value, range 0 to 255
FrameReadout            = FULL         // FULL,PARTIAL
ExpTime                 = 72           // Unit - Milisecond,
                                       // Range - 1 mili sec to 9.5 hours
ROI_X1                  = 0000         // ROI : Region of Interest coordinates
ROI_Y1                  = 0000         //
ROI_X2                  = 0000
ROI_Y2                  = 0000
NResets                 = 1            // Number of RESET frames at the
                                       // beginning of exposure
NReads                  = 1            // Number of READ frames
NDrops                  = 0            // Number of Drop frames per
                                       // Group (applicable for URG sampling
                                       // scheme only)
NGroups                 = 0            // Numbers of Groups per Ramp
NRamps                  = 0            // Number of Exposures
Sampling                = URG          // URG,FS
//
IFPACMode               = 1            //1: Single detector/2: Multi detector
SendADCData             = YES          //YES/NO
//
```