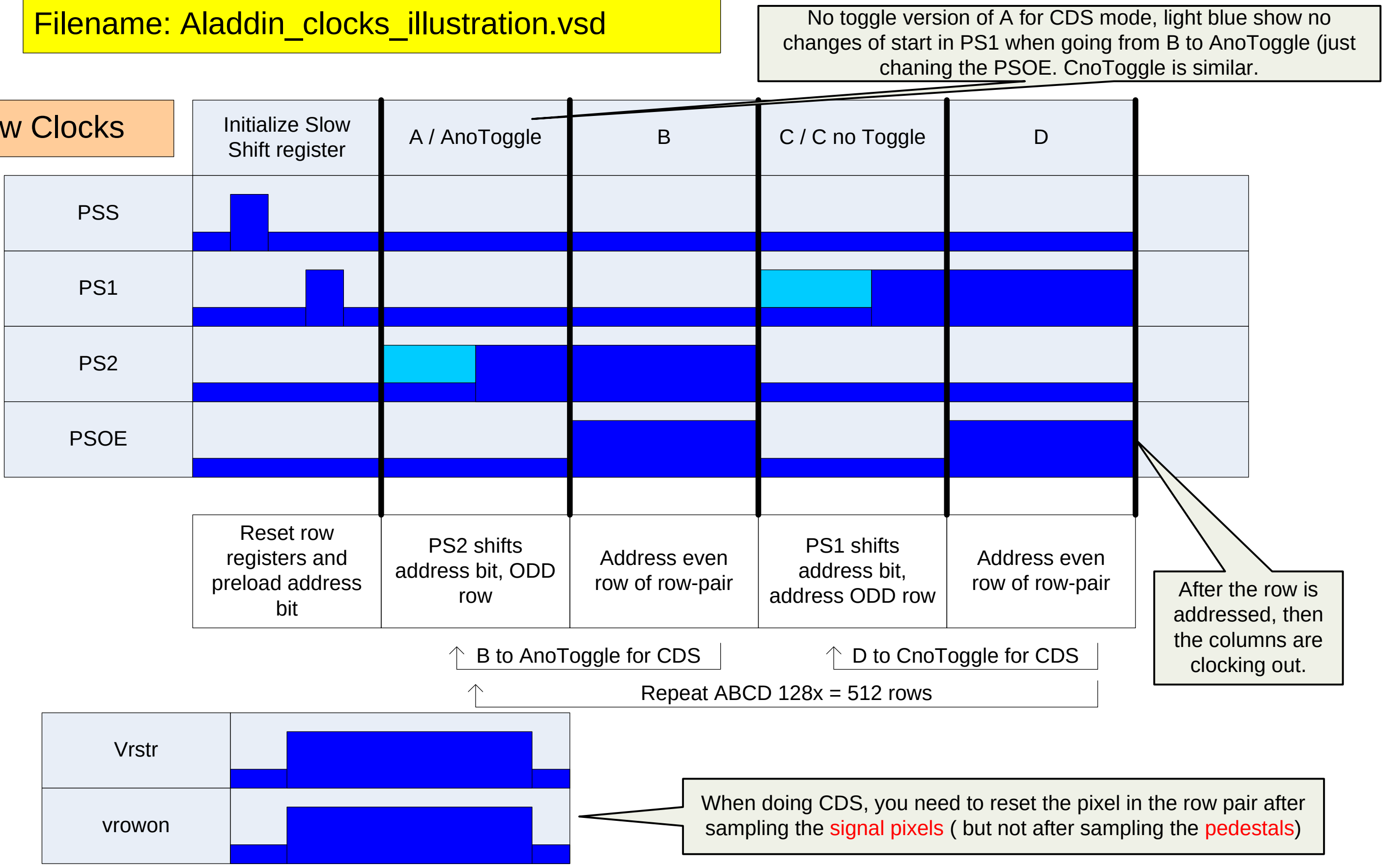
Spex Aladdin Clocks Illustration
Last Edit: 4/21/2010
Filename: Aladdin_clocks_illustration.vsd

Row Clocks

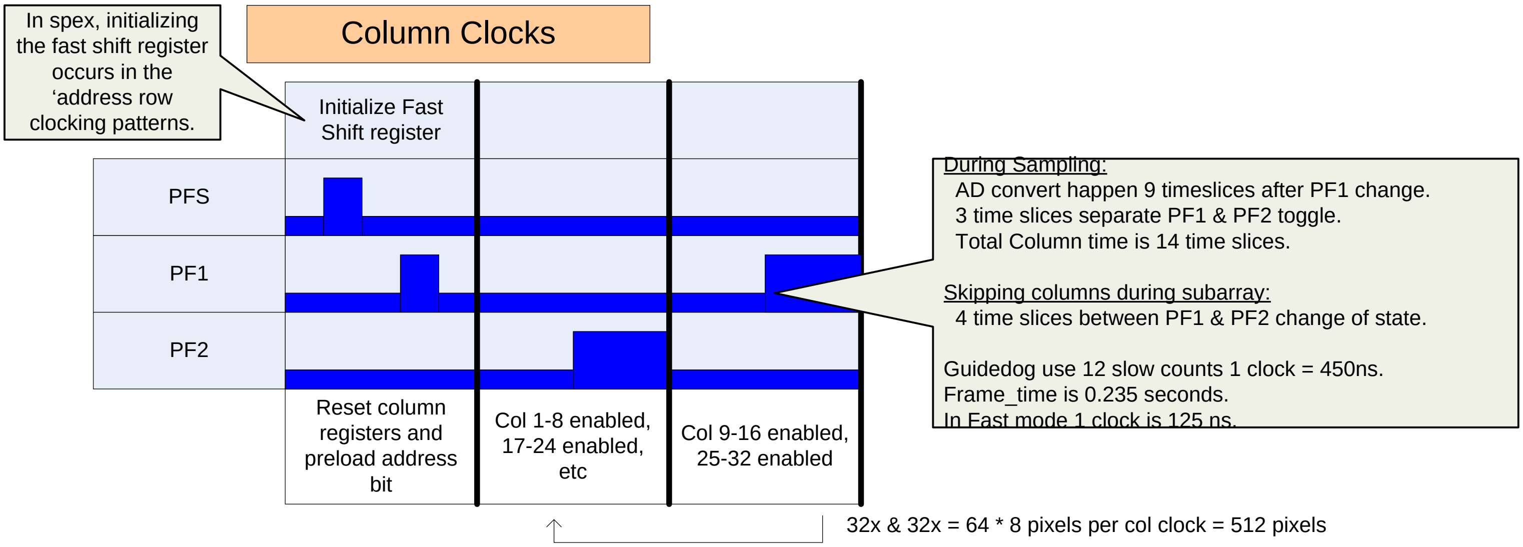

Initialize Slow Shift register
A / AnoToggle
B
C / C no Toggle
D
PSS
PS1
PS2
PSOE
Reset row registers and preload address bit
PS2 shifts address bit, ODD row
Address even row of row-pair
PS1 shifts address bit, address ODD row
Address even row of row-pair
B to AnoToggle for CDS
D to CnoToggle for CDS
Repeat ABCD 128x = 512 rows
Vrstr
vrowon

No toggle version of A for CDS mode, light blue show no changes of start in PS1 when going from B to AnoToggle (just chaning the PSOE. CnoToggle is similar.

After the row is addressed, then the columns are clocking out.

When doing CDS, you need to reset the pixel in the row pair after sampling the signal pixels (but not after sampling the pedestals)

Column Clocks


Initialize Fast Shift register
PFS
PF1
PF2
Reset column registers and preload address bit
Col 1-8 enabled, 17-24 enabled, etc
Col 9-16 enabled, 25-32 enabled
32x & 32x = 64 * 8 pixels per col clock = 512 pixels

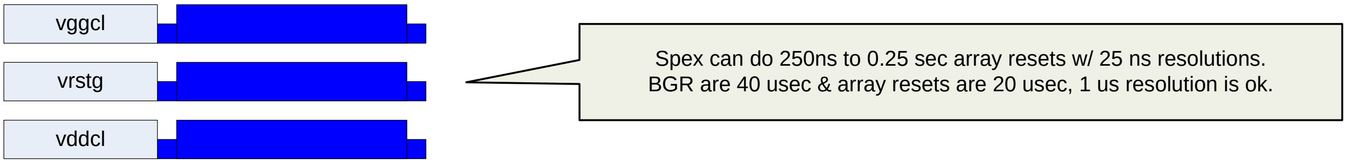
In spex, initializing the fast shift register occurs in the ‘address row clocking patterns.

During Sampling:
AD convert happen 9 timeslices after PF1 change.
3 time slices separate PF1 & PF2 toggle.
Total Column time is 14 time slices.

Skipping columns during subarray:
4 time slices between PF1 & PF2 change of state.

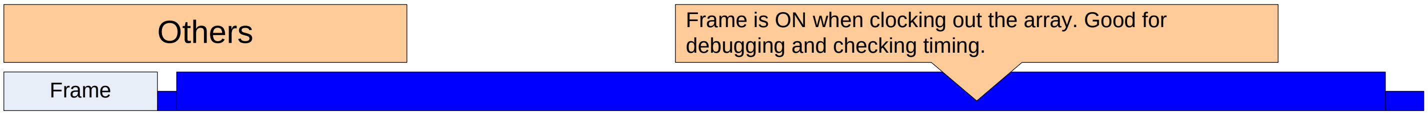
Guidedog use 12 slow counts 1 clock = 450ns.
Frame_time is 0.235 seconds.
In Fast mode 1 clock is 125 ns.

Global Resets



Spex can do 250ns to 0.25 sec array resets w/ 25 ns resolutions.
BGR are 40 usec & array resets are 20 usec, 1 us resolution is ok.

Others



Frame is ON when clocking out the array. Good for debugging and checking timing.

Column Mux Register

Current Spex row timing:
Guidedog’s FASTMODE is 125 ns TimeSlices
GuideDog’s default is 12 slowcnt or 450 ns TimeSlices
When sampling Column, we used 35 TS: 9 TS before converts, and 3 between PF1/PF2 toggle.

Can we used slowcnt with the SGIR pattern generator? Yes, based on 35 TimeSlice, we can do this:

USING SGIR SLOWCNT: ns = 40 + (20*slowcnt)					
Slowcnt	slowcnt_ns		Pattern		Frame_Rate ms
4	120		4200		68.8128
20	440		15400		252.3136

Clock the row and convert. Aladdin we group 2 column in a pattern generator, 16 pixels (8channell * 2):

		0	1	2	3	4	5	6	7	8	
PG3	b0										
	b1										
	b2 PFS										
PG4	b0 - PF1						■	■	■		
	b1- PF2		■	■	■						
	b2 - vclamp										
	b3 -ADCTRIG			■				■			
Spex TS		3	9	1	4	3	9	1	4	1	35
To get similar Timing we can do this:											
SGIR		3	9	2	2	3	9	2	2	1	33
		Yellow block timing determine by ADC parameters,									
		This example uses 2 or 220 ns (2 samples, 1 channel)									

Yellow assumes 1 ch, 2 sample or 110ns*2 = 220. That’s equivalent to 2 120 ns timeslices (240ns)

That means, slowcnt and numsample defines the timing.

Clock a row with no converts: to clock quickly over row (used for subarrays)

		0	1	2	3	4	5	6	7	8	
PG3	b0										
	b1										
	b2 PFS										
PG4	b0 - PF1						■	■	■		
	b1- PF2		■	■	■						
	b2 - vclamp										
	b3 -ADCTRIG										
Spex TS		4	4			4	4			0	16
To get similar Timing we can do this:											
SGIR		4	2	1	1	4	2	1	1	1	17

To initialize fast shift register, we need to reprogram PG3 / 4. This will happen 1 once evey row

	b2 PFS		■								
PG4	b0 - PF1				■						
	b1- PF2										
	b2 - vclamp										
	b3 -ADCTRIG										
Spex TS		4	4	4	4	4	0	0	0	0	20
To get similar Timing in SGIR we can do this:											
SGIR		4	4	4	4	1	1	1	1	1	21

Clocking Rows

There are 4 signal used to control the Row MUX: PSS, PS1, PS2, PSOE
PS1, and PS2 are used to march the address bit on the MUX. This address bit address a row-pair: AB or CD.
Then the PSOE is used to select one of the row pairs.
It is best to used DIGOUT for the row mux signal. (rather that the parallel pattern generator)

PSS		▨						
PS1				▨				
PS2								
PSOE								
							Total	
spex TS	1	4	4	4	4		17	
SGIR TS	1	4	4	4	4			

Here we intialize the row mux, by toggling SS(slow sync) and the S1. SGIR will do the same using DIGIOUT and Delays.

	Row A			Row B		Row C		Row D						
PSS														
PS1						▨	▨	▨	▨					
PS2		▨	▨	▨	▨									
PSOE				▨	▨			▨	▨					
			*		*		*		*		* = columns are readout here.			
spex TS	4	18		18		18		18			Spex used 18 TS because the			
											fast MUX are initialize here.			
SGIR TS	4	8		8		8		8			Can be as fast as 4 TS.			

The above table show how we can used DIGOUT to march the Row Address Bit down the Row (slow) MUX. For SGIR we will not do CDS mode (just fowler sampling), so the CDS scheme will not be illustrated, but can easily be accomplished.

Vrstr		▨				
vrowon		▨				
Spex TS	2	16	2			

Just for completeness, the Vrstr, and vrowon are used to reset the pixels in the row pair during CDS mode. The TS used in spex is displayed here.